



The `cgeostat` Software for Analyzing Complex-Valued Random Fields

Sandra De Iaco
Università del Salento

Abstract

Given a vectorial data set in two dimensions, a representation on a complex domain is often convenient. This representation is rarely considered in geostatistics, although interesting applications can be found in environmental sciences and meteorology (e.g., for wind fields). In such a case, some computational difficulties are related to the lack of software for estimating and modeling a complex covariance function, for predicting complex variables as well as for representing the output results. In this paper, the new Fortran software `cgeostat` for geostatistical analysis of complex-valued random fields is presented and an application is demonstrated.

Keywords: complex variables, bivariate random fields, complex representation, complex covariance function, prediction algorithm, **GSLib**.

1. Introduction

Complex-valued random fields are often appropriate to model vectorial data with two components: This kind of data can be found in many areas of research, such as in hydrology and meteorology (i.e., in wind or stream studies). The formalism of complex-valued random fields can reflect the specific characteristics of the components of a vectorial random field: These components are associated with a physical phenomenon described by homogeneous quantities, characterized by the same unit of measure, such as wind velocity, force, electric or magnetic field, which are available at the same spatial points over the domain (De Iaco, Maggio, Palma, and Posa 2012). In this case, the corresponding realization of a vectorial field in two dimensions is an expression of a single entity, i.e., a complex number, where the decomposition in modulus and angle is natural and has a physical interpretation. Thus, a compact and unified formalism has to be suitably adopted for this kind of data. In other terms, the use of a complex formalism and consequently of complex kriging for prediction purposes is not an

arbitrary choice, but it is strictly motivated by the nature of the phenomenon. Nevertheless, geostatistical applications are usually concerned with scalar variables or only with the magnitude of vectorial variables, such as the norm of the wind vector (Haslett and Raftery 1989; Cressie and Huang 1999).

The specific formalism of complex-valued random fields for vectorial data with two components, as well as various formulations of the predictor for bivariate random fields with a representation on a complex domain, were proposed by Lajaunie and Béjaoui (1991) and Wackernagel (2003). In particular, Lajaunie and Béjaoui (1991) applied the complex kriging and spent some effort to describe valid covariance functions. They also compared the complex kriging to simple kriging or direct cokriging of the two components of the vectorial variable. Some other contributions concern admissible complex covariance models, such as the one given by De Iaco, Palma, and Posa (2003), where the complex covariance models are generated by starting from a real covariance model and its spectral representation, and by introducing a shifting factor in order to translate the spectral density function.

Even if a lot can still be done from a theoretical point of view, computational advances in complex geostatistical analysis are also in great demand. In particular, for a wider use of the geostatistical theory of complex-valued random fields, some computational aspects on complex prediction should be developed.

In the literature, there exist various packages which perform univariate and multivariate geostatistical analysis in two and three dimensions, such as **Geo-EAS** (Englund 1991), **GEOSAN** (Dutter 1996), the specific routine for assessing the fit of a variogram model (Barry 1996), the *Geostatistical Software Library* **GSLib** (Deutsch and Journel 1998) and its version for Windows **WinGSLIB** (Staios 2001), **SGeMS** (Remy, Boucher, and Wu 2009), as well as the commercial software **ISATIS** (Geovariances 2015). Moreover, the R programming language (R Core Team 2017) has various packages devoted to geostatistics, such as the **gstat** package for geostatistical modeling, prediction and simulation (Pebesma and Wesseling 1998; Pebesma 2004; Pebesma and Bivand 2005), **geoR** (Ribeiro Jr. and Diggle 2001; Diggle and Ribeiro Jr. 2007), which contains functions for model-based geostatistics, as well as the **geospt** (Melo, Santacruz, and Melo 2015), **RandomFields** (Schlather, Malinowski, Menck, Oesting, and Storkorb 2015) and **RGeostats** (Renard, Desassis, Beucher, Ors, and Laporte 2014) packages, or ad-hoc routines for graphical interface in the ecological modeling software **Bio7** (Austenfeld and Beyschlag 2012) or for exploratory spatial data analysis (Laurent, Ruiz-Gazen, and Thomas-Agnan 2012). Other contributions concern specialized routines and packages for spatio-temporal analysis (De Cesare, Myers, and Posa 2002; De Iaco, Myers, Palma, and Posa 2010; Pebesma 2012; De Iaco and Posa 2012; Gabriel, Rowlingson, and Diggle 2013). However, none of the above mentioned packages provides tools for estimating and modeling the real and imaginary parts of a complex covariance function as well as for predicting complex-valued random fields.

In this paper, after introducing the framework of complex-valued random fields, a discussion on complex kriging and some details on complex covariance models are presented (Section 2). Then, the new software **cgeostat**, to be applied for spatial analysis of phenomena with a representation on a complex domain, is proposed (Section 3). This is a software package which has been written in Fortran by using the *Geostatistical Software Library* **GSLib**, given in the version called **GSLIB77**. Note that these **GSLib** source codes can be freely downloaded from the website <http://www.staios.com/Quick/gslib.html> and their availability allows the basic algorithms to be used for more advanced customized programs. The **cgeostat** software package presented in this paper includes the program **ckriging** for predicting complex vari-

ables and some other programs for: representing complex data (`clommap`), computing and plotting components of a sample complex covariance function (`ccovaplt`), fitting and computing components of a complex covariance model (`cfitting` and `ccovamodel`). The program `ckriging` provides various options for prediction, such as complex kriging for a grid of points or blocks, cross-validation and jackknife; moreover the user can choose between simple kriging, if the expected value is known, and ordinary kriging, if the expected value is unknown. In Section 4, the new software is applied to study a complex-valued random variable and to provide predictions of the same variable at some unsampled locations on the basis of the available data. Note that cross-validation and jackknife options are also considered and some descriptive and inferential statistics regarding prediction performance are discussed.

2. Theoretical framework on complex-valued random fields

Let U and V be two real-valued random fields, which represent the components of the complex-valued random field, defined as follows

$$W(\mathbf{u}) = U(\mathbf{u}) + i V(\mathbf{u}), \quad \mathbf{u} \in \mathbb{R}^d, \quad (1)$$

where $i = \sqrt{-1}$ and d is the dimension of the spatial domain ($d \in \mathbb{N}_+$, $d \leq 3$). The random field W is stationary of second order, if its expected value, denoted with m , exists and is constant for any \mathbf{u} over the domain and the covariance function C depends on the lag separation vector \mathbf{h} , for any two points \mathbf{u} and $\mathbf{u} + \mathbf{h}$, that is

$$C(\mathbf{h}) = \mathbf{E}[(W(\mathbf{u}) - m)(\overline{W(\mathbf{u} + \mathbf{h}) - m})],$$

where the complex conjugation is denoted with $\bar{\cdot}$. The above complex covariance function can be explicitly written as,

$$C(\mathbf{h}) = C^{Re}(\mathbf{h}) + iC^{Im}(\mathbf{h}), \quad (2)$$

where the real part is defined by the sum of the covariances of U and V , i.e., $C_U(\mathbf{h})$ and $C_V(\mathbf{h})$, respectively, while the imaginary part is given by the difference between the related cross-covariances $C_{VU}(\mathbf{h})$ and $C_{UV}(\mathbf{h})$,

$$C^{Re}(\mathbf{h}) = C_U(\mathbf{h}) + C_V(\mathbf{h}), \quad C^{Im}(\mathbf{h}) = C_{VU}(\mathbf{h}) - C_{UV}(\mathbf{h}). \quad (3)$$

Note that the real part of (2) is an even function and, because of the convexity property (Chilés and Delfiner 1999), it is a covariance function, whereas the imaginary part of (2) is an odd function and is equal to zero in presence of isotropy and cannot be considered as a covariance function.

Moreover, for a complex-valued random field W , which is stationary of second order, the positive definiteness condition of C is given by:

$$\sum_{i=1}^k \sum_{j=1}^k C(\mathbf{u}_i - \mathbf{u}_j) a_i \bar{a}_j \geq 0,$$

for any $n \in \mathbb{N}_+$ and any choice of points $\mathbf{u}_i \in \mathbb{R}^d$, and $a_i \in \mathbb{C}$, $i = 1, \dots, n$.

2.1. Some details on complex covariance models

For prediction purposes, it is necessary to model C^{Re} (defined as the sum of the covariances C_U and C_V) and C^{Im} (defined as the difference of the two cross-covariances C_{VU} and C_{UV}) in a consistent way such that C is admissible, i.e., that it is a complex covariance function. In particular, [De Iaco *et al.* \(2003\)](#) showed that

$$C(\mathbf{h}; \mathbf{c}) = \cos(\mathbf{h} \cdot \mathbf{c})\tilde{C}(\mathbf{h}) + i \sin(\mathbf{h} \cdot \mathbf{c})\tilde{C}(\mathbf{h}), \quad (4)$$

is a family of complex covariance functions, where $\tilde{C}(\mathbf{h})$ is a real covariance function and $\mathbf{c} \in \mathbb{R}^d, \mathbf{c} \neq \mathbf{0}$. In other words, given a real covariance function $\tilde{C}(\mathbf{h})$ and the corresponding even spectral density function $f(\mathbf{s})$, complex covariance models can be obtained by shifting the function $f(\mathbf{s})$, i.e., given a vector $\mathbf{c} \neq \mathbf{0}$, the function $f(\mathbf{s} - \mathbf{c})$ is a not-even spectral density function $\forall \mathbf{c} \in \mathbb{R}^d, \mathbf{c} \neq \mathbf{0}$, hence, the spectral representation based on this translated function can be used to generate a complex covariance model.

In this paper, the above complex covariance models are used for modeling the correlation structure of the variables under study, since:

- the family of complex covariance models in (4) is generated by starting from real covariance functions, thus various examples of complex covariance functions can be easily obtained by using such a model;
- the fitting procedure is shown to be flexible enough.

As described in [De Iaco, Posa, and Palma \(2013\)](#), the family of complex covariance functions, given in (4), can be fitted by using an easy step by step procedure. After estimating the covariance functions for U and V and the corresponding cross-covariances for different lags and directions of the spatial domain (step I), the presence of anisotropy with type and direction of maximum continuity can be detected (step II). Then the components of the complex covariance function in (2), i.e., the sum of the sample direct covariances and the difference of the sample cross-covariances for relevant directions, have to be computed (step III). At last, the shifting vector \mathbf{c} has to be estimated, for example through least squares regression (step IV), and an appropriate model for \tilde{C} can be easily chosen by inspecting the real part of the sample complex covariance function (step V).

2.2. Prediction of complex random fields

Let W be a second order stationary complex-valued random field, defined as in (1). Prediction of W at a location \mathbf{u}_0 can be performed through different forms of kriging, which can be considered as an extension of the ones known for random fields on a real domain: The simple complex kriging and the ordinary complex kriging can be used depending on whether the expected value of W is known or not, as described below.

Let the predictor of the complex random variable $W(\mathbf{u}_0)$ at an unsampled location \mathbf{u}_0 be

$$\widehat{W}_K(\mathbf{u}_0) = \sum_{\alpha=1}^n \omega_{\alpha} W(\mathbf{u}_{\alpha}) = \boldsymbol{\omega}^{\top} \mathbf{W}, \quad (5)$$

where \mathbf{W} represents the vector $[W(\mathbf{u}_1), W(\mathbf{u}_2), \dots, W(\mathbf{u}_n)]^{\top}$ of the complex random variables $W(\mathbf{u}_{\alpha})$ at the sampled locations \mathbf{u}_{α} and $\boldsymbol{\omega}$ corresponds to the vector of the complex weights

$[\omega_1, \omega_2, \dots, \omega_n]^\top$. In terms of the real and imaginary parts, predictor (5) can be written as

$$\begin{aligned}\widehat{W}_K(\mathbf{u}_0) &= \sum_{\alpha=1}^n [\omega_\alpha^{Re} U(\mathbf{u}_\alpha) - \omega_\alpha^{Im} V(\mathbf{u}_\alpha)] + i[\omega_\alpha^{Re} V(\mathbf{u}_\alpha) + \omega_\alpha^{Im} U(\mathbf{u}_\alpha)] \\ &= \mathbf{U}^\top \boldsymbol{\omega}^{Re} - \mathbf{V}^\top \boldsymbol{\omega}^{Im} + i(\mathbf{U}^\top \boldsymbol{\omega}^{Im} + \mathbf{V}^\top \boldsymbol{\omega}^{Re}),\end{aligned}$$

where $\boldsymbol{\omega}^{Re}$ and $\boldsymbol{\omega}^{Im}$ are the vectors of the real and imaginary parts of the vector $\boldsymbol{\omega}$, while $\mathbf{U} = [U(\mathbf{u}_1), U(\mathbf{u}_2), \dots, U(\mathbf{u}_n)]^\top$ and $\mathbf{V} = [V(\mathbf{u}_1), V(\mathbf{u}_2), \dots, V(\mathbf{u}_n)]^\top$ are, respectively, the vectors of the real and the imaginary part of \mathbf{W} . It is easy to show that $\widehat{W}_K(\mathbf{u}_0)$ is unbiased if and only if $\sum_{\alpha=1}^n \omega_\alpha = 1$, which implies that

$$\sum_{\alpha=1}^n \omega_\alpha^{Re} = 1, \quad \sum_{\alpha=1}^n \omega_\alpha^{Im} = 0.$$

Under the unbiasedness condition, if it is required that the error variance is minimum, then the following complex ordinary kriging system, based on $(2n + 2)$ equations and $(2n + 2)$ parameters, is obtained

$$\begin{bmatrix} \mathbf{C}^{Re} & \mathbf{C}^{Im} & \mathbf{1} & \mathbf{0} \\ \mathbf{C}^{Im\top} & \mathbf{C}^{Re} & \mathbf{0} & \mathbf{1} \\ \mathbf{1}^\top & \mathbf{0}^\top & 0 & 0 \\ \mathbf{0}^\top & \mathbf{1}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}^{Re} \\ \boldsymbol{\omega}^{Im} \\ M^{Re} \\ M^{Im} \end{bmatrix} = \begin{bmatrix} \mathbf{c}^{Re} \\ \mathbf{c}^{Im} \\ 1 \\ 0 \end{bmatrix}, \quad (6)$$

where $\mathbf{C}^{Re} = [C^{Re}(\mathbf{u}_\alpha - \mathbf{u}_\beta)]_{(n \times n)}$, $\mathbf{C}^{Im} = [C^{Im}(\mathbf{u}_\alpha - \mathbf{u}_\beta)]_{(n \times n)}$, with $\alpha, \beta = 1, 2, \dots, n$, $\mathbf{1} = [1]_{(n \times 1)}$, $\mathbf{0} = [0]_{(n \times 1)}$, $\mathbf{c}^{Re} = [C^{Re}(\mathbf{u}_0 - \mathbf{u}_\alpha)]_{(n \times 1)}$ and $\mathbf{c}^{Im} = [C^{Im}(\mathbf{u}_0 - \mathbf{u}_\alpha)]_{(n \times 1)}$, with $\alpha = 1, 2, \dots, n$, while M^{Re} and M^{Im} are the components of the complex Lagrange multiplier $M = M^{Re} + iM^{Im}$.

If the complex mean function $m(\mathbf{u}) = m_U(\mathbf{u}) + im_V(\mathbf{u})$, with $m_U(\mathbf{u}) = E[U(\mathbf{u})]$ and $m_V(\mathbf{u}) = E[V(\mathbf{u})]$, is known or separately estimated over the study area, the residuals U' and V' of the real and imaginary parts can be obtained as follows

$$U'(\mathbf{u}) = U(\mathbf{u}) - m_U(\mathbf{u}), \quad V'(\mathbf{u}) = V(\mathbf{u}) - m_V(\mathbf{u}).$$

Then, the residual complex random field $W'(\mathbf{u}) = W(\mathbf{u}) - m(\mathbf{u})$ can be defined as

$$W'(\mathbf{u}) = U'(\mathbf{u}) + iV'(\mathbf{u}).$$

Hence, complex simple kriging of W' , based on $2n$ equations and $2n$ parameters, can be performed by solving the following system

$$\begin{bmatrix} \mathbf{C}^{Re} & \mathbf{C}^{Im} \\ \mathbf{C}^{Im\top} & \mathbf{C}^{Re} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega}'^{Re} \\ \boldsymbol{\omega}'^{Im} \end{bmatrix} = \begin{bmatrix} \mathbf{c}^{Re} \\ \mathbf{c}^{Im} \end{bmatrix}, \quad (7)$$

where $\mathbf{C}^{Re} = [C^{Re}(\mathbf{u}_\alpha - \mathbf{u}_\beta)]_{(n \times n)}$ and $\mathbf{C}^{Im} = [C^{Im}(\mathbf{u}_\alpha - \mathbf{u}_\beta)]_{(n \times n)}$, with $\alpha, \beta = 1, 2, \dots, n$, while $\mathbf{c}^{Re} = [C^{Re}(\mathbf{u}_0 - \mathbf{u}_\alpha)]_{(n \times 1)}$ and $\mathbf{c}^{Im} = [C^{Im}(\mathbf{u}_0 - \mathbf{u}_\alpha)]_{(n \times 1)}$, with $\alpha = 1, 2, \dots, n$. Note

that the above components of the complex covariance function C are related to the variable W' .

As will be clarified in the following section, the complex covariance model in (4) and the above systems (6) and (7) have been suitably implemented in the new software **cgeostat**.

Remarks

Some general aspects on complex formalism for bivariate data and a theoretical comparison of complex kriging with respect to other techniques (such as separate kriging and complex cokriging) were proposed by [Wackernagel \(2003\)](#). Thus, any empirical comparison would reflect the already known theoretical results, as highlighted in [De Iaco *et al.* \(2013\)](#). As pointed out in the introduction, in addition to these aspects, the difference between complex formalism and other (univariate or multivariate) alternatives is not only formal, but conceptual, since any realization of a vectorial field in two dimensions (such as a wind field, streams, an electric or magnetic field) is expression of a single entity, i.e., a complex-valued function; for this reason, the compact and unified complex formalism is suitably adapted for this kind of data. From a theoretical and practical point of view, the advantages of using complex formalism, whenever the phenomenon requires this formalism, are described below:

- Complex kriging is adequate for predicting vectorial data if the components, characterized by homogeneous quantities, are correlated and the decomposition in modulus and angle is physically meaningful. An interesting discussion on the rotation and the scaling effects produced, to vectorial data, by the complex kriging weights, can also be found in [De Iaco *et al.* \(2013\)](#).
- The complex covariance completely characterizes the correlation structure of vectorial data in two dimensions; this enables the computation of spatial correlations in an easier and more flexible way compared to bivariate techniques.
- Using complex kriging simplifies the prediction step, since the complex kriging system depends on $2n$ parameters (where n is the number of available data), while the “complex cokriging” system, which is equivalent to the well-known formalism used for the cokriging predictions of a real-valued random vector or multivariate random field of two components (bivariate cokriging), measured at the same sample locations (isotopic case), depends on $4n$ parameters. If bivariate cokriging is used, it is not enough to have a model for the sum and the difference of direct and cross covariances, but a valid model for every direct and cross covariance function is also required.

3. Software for complex random fields analysis (tutorial)

The principal problem when analyzing bivariate data, with a reasonable representation on a complex domain, is the lack of software. Thus, customized **GSLib** ([Deutsch and Journel 1998](#)) Fortran programs for complex data representation (`clocmap`), estimation and plotting components of a sample complex covariance function (`ccovaplt`), fitting and computing a complex covariance model (`cfitting` and `ccovamodel`), complex kriging prediction (`ckriging`) have been implemented and are presented in this paper.

Note that the parameters (such as variables to be used, input/output files, options) have to be set by the user through an ad-hoc file, named parameter file (see Appendix A for details). These Fortran programs, together with the **GSLib** library (see Appendix B for details on dependencies), and the corresponding executable programs are freely available, moreover a directory with the parameter files, the input and output files used for the example given in this article are also included.

3.1. Program `clocmap`

The program `clocmap` can be used to plot a location map for a complex-valued variable, measured at some spatial points over a two-dimensional domain. The data values for the two components U and V are plotted by using the usual representation on a complex domain, that is each pair of values is illustrated with an arrow whose modulus and direction depend on U and V . If the program `clocmap` is applied to represent the complex kriging estimates for U and V , then their error variance (with the option to apply some transformation, such as the square root) can be also illustrated with a circle whose diameter is proportional to the magnitude of the error variance. Moreover, the overlapping of the location map of the complex data used to compute the estimates (called conditioning data) can be also considered. Different colors for the location map of the estimates and the location map of the conditioning data can be chosen. Note that if the data set is referred to a 3D domain, a 2D representation can be obtained by using an option in the program `clocmap`, where the user can specify the 2D section, for a given coordinate z , to be plotted; in this way, the program can be used to produce different profiles of the data corresponding to various sections.

Parameter, input and output files

In the following, the input and output files of the program `clocmap` and the associated parameter file are described.

There are two input files:

- The first file is mandatory and represents the data file with at least the two components of the complex data (and if necessary a third variable) for each location, in **Geo-EAS** format.
- The second one is optional and represents the data file with the conditioning data (sample data for the two components, which have been used for structural analysis), in **Geo-EAS** format; indeed, this is required only when the overlay option is set equal to 1.

These files are ASCII files (or text files), their extension is usually `.dat` and their names can be arbitrarily fixed in the parameter file.

Regarding the output, the program `clocmap` creates a postscript file.

An example of a parameter file for the program `clocmap` is given below.

```
Parameters for clocmap  
*****
```

START OF PARAMETERS:

```

data19x24.txt          \ file with data
1  2  0  4  5  0      \ columns for coords X, Y, Z & vars U, V, E
-9990.  1.0e21        \ trimming limits for U
-9990.  1.0e21        \ trimming limits for V
-9990.  1.0e21        \ trimming limits for E
0                      \ option overlay: 0 = no, 1 = yes
XXXX.txt              \ file with conditioning data
1  2  0  4  5          \ columns for coords X, Y, Z & vars U, V
1                      \ color code for conditioning data U, V
clocmap19x24.ps        \ file for postscript output
50.0   290.            \ xmn, xmx
35.0   340.            \ ymn, ymx
0.0                      \ z selection
8                      \ color code for vector U,V
0                      \ scaling: 0 = arithmetic, 1 = log10 (|u|>1, |v|>1)
0.01  12.0  3  10     \ modulus scale & size: dmin, dmax, smin, smax
0                      \ 0 = no labels, 1 = label each location
8                      \ color code for variable E
1                      \ transform var.E: 0 = none, 1 = sqrt, 2 = log10
0.5                      \ label size of var.E: 0.1 = sml, 1 = reg, 10 = big
Location map of complex data \ title

```

In particular, apart from the parameters which are similar to the ones that are used in **GSLib** (Deutsch and Journel 1998), it is worth noting that:

- the parameter that specifies the color code might take an integer number from 1 to 8 (1 = red, 2 = yellow, 3 = green, 4 = light blue, 5 = dark blue, 6 = violet, 7 = white, 8 = black);
- the scaling parameter might be fixed equal to 0, if no scaling transformation is applied to the components U and V , while it is fixed equal to 1 if the logarithmic transform is applied;
- regarding the parameters that specify the size of the symbol according to the modulus scale, if data minimum (**dmin**) is set equal or greater than data maximum (**dmax**), minimum and maximum are found from the data. Moreover, if minimum size (**smin**) is set equal or greater than maximum size (**smax**), minimum size and maximum size are set equal to the data minimum and maximum;
- if the overlay option is set equal to 1, then the parameters regarding color, scale transformation and symbol size for the additional variable (usually the error variance, if available) can be appropriately fixed.

Run clocmap

If the parameter file has been appropriately set, the user can run the program `clocmap`. After typing the name of the parameter file, the program starts working.

The output is saved as a postscript file, whose name is defined in the parameter file. The ghostscript program would be a useful supplement for visualizing the graphical output.

Besides the error alerts which are common to all programs, specific error messages for the program `ccovaplt` may be issued:

- to point out if the columns indicated for the coordinates and the components U and V are missing or not valid,
- to detect if the number of data to be stored exceeds the maximum limit (which is equal to 10000).

3.2. Program `ccovaplt`

After computing the direct and cross directional sample covariance functions by using the available program `gamvm` of the **GSLib** library, it is necessary to determine and plot the components of the sample complex covariance function along different directions. Then the program `ccovaplt` can be applied.

Parameter, input and output files

In the following, the input and output files of the program `ccovaplt` and the associated parameter file are described.

First of all, the input files might be:

- the direct and cross sample covariance functions, determined for different directions, in the format produced by the output of the program `gamvm` of the **GSLib** library;
- the components of the complex covariance model, in the format produced by the output of the program `ccovamodel` (details of this program are given in Section 3.4).

These files are ASCII files (or text files) and their names can be arbitrarily fixed.

The output of this program is a postscript file and its name is also specified in the parameter file.

The parameter file required by the program `ccovaplt` is given below.

Parameters for ccovaplt

START OF PARAMETERS:

```

plt_ccova_SE.ps      \ file for postscript output
4                   \ number of covariances to plot
0.0    100.0         \ distance limits (from data if max < min)
-10.0   30.0         \ covariance limits (from data if max < min)
Components of C -SE dir \ title for covariance function
2                   \ comp. on data to plot: none = 1, sum = 2, dif = 3
covaU.txt           \ 1 file with covariance data
covaV.txt           \ 1 file with covariance data (only for type = 2, 3)

```

```

3    1    1    0    4    \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
3                                     \ comp. on data to plot: none = 1, sum = 2, dif = 3
crosscovaVU.txt                       \ 1 file with covariance data
crosscovaUV.txt                       \ 1 file with covariance data (only for type = 2, 3)
3    1    1    0    5    \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
2                                     \ comp. on data to plot: none = 1, sum = 2, dif = 3
ccovamodel_real.var                   \ 1 file with covariance data
XXX.txt                               \ 1 file with covariance data (only for type = 2, 3)
3    0    0    1    4    \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
1                                     \ comp. on data to plot: none = 1, sum = 2, dif = 3
ccovamodel_im.var                     \ 1 file with covariance data
XXX.txt                               \ 1 file with covariance data (only for type = 2, 3)
3    1    0    1    5    \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr

```

In particular, apart from the comments given in the parameter file for the first 5 lines which clearly explain what is required, it is worth noting that

- the parameter that specifies the computations on the data to plot, can be fixed:
 - equal to 1, if the covariance values provided in the subsequent input file, are plotted without any computations;
 - equal to 2, if the sum (lag by lag) of covariance values provided in the two subsequent input files, has to be determined and plotted (this is convenient to obtain the real component of the complex covariance);
 - equal to 3, if the difference (lag by lag) of covariances provided in two subsequent input files, has to be determined and plotted (this is convenient to obtain the imaginary component of the complex covariance).
- for each covariance function to plot, the following parameters have to be set:
 - the serial number that identifies the covariance to plot, selected among the covariances recorded (collected) in the indicated “file with covariance data”;
 - the style options for the line: dashed line (integer numbers from 1 to 10, according to the desired length of the dash), point between dashes (1 = yes, 0 = no), continuous line (1 = yes, 0 = no) and color (integer numbers from 1 to 8, according to the color codes which are 1 = red, 2 = yellow, 3 = green, 4 = light blue, 5 = dark blue, 6 = violet, 7 = white, 8 = black).

Run ccovaplt

If the parameter file has been suitably specified, the user can run the program `ccovaplt` and type the name of the parameter file.

When the program has terminated, the output is saved as a postscript file, whose name is defined in the parameter file.

Moreover, the program `ccovaplt` produces a specific error message to acknowledge if the number of covariances exceeds the maximum value (equal to 10).

3.3. Program `ccfitting`

After computing and representing the components of the empirical complex covariance function along different directions, it is useful to estimate the translating vector \mathbf{c} through non-linear least squares. The program `ccfitting` can be applied to this end.

Thus, the input files are the direct and cross sample covariance functions, determined for different directions, in the format produced by the output of the program `gamvm` of the **GSLib** library.

The output of this program is an ASCII file (or text file) and its name is arbitrarily defined in the parameter file.

The parameter file required by the program `ccfitting` is given below.

Parameters for ccfitting

START OF PARAMETERS:

```

estimate.txt      \ file for output
2                 \ number of direction to consider (>= 2)
45      0.0       \ 1st direction (degree clockwise from North) dip
3                 \ number of lags to be considered
covaU.txt         \ 1 file with directional covariance for U
covaV.txt         \ 1 file with directional covariance for V
2                 \ position of the covariance in the files
crosscovaVU.txt   \ 1 file with directional cross-covariance for VU
crosscovaUV.txt   \ 1 file with directional cross-covariance for UV
2                 \ position of the covariance in the files
135    0.0       \ 2nd direction (degree clockwise from North) dip
3                 \ number of lags to be considered
covaU.txt         \ 1 file with directional covariance for U
covaV.txt         \ 1 file with directional covariance for V
3                 \ position of the covariance in the files
crosscovaVU.txt   \ 1 file with directional cross-covariance for VU
crosscovaUV.txt   \ 1 file with directional cross-covariance for UV
3                 \ position of the covariance in the files

```

Note that the comments given in the parameter file are sufficiently informative about the required parameters.

3.4. Program `ccovamodel`

Another program is `ccovamodel` which calculates the components of a complex covariance model at certain lags and writes the result in a format compatible with `ccovaplt`. The parameter file required by this program is illustrated below.

Parameters for ccovamodel

START OF PARAMETERS:

```

ccovamodel_real.var          \ file for covariance output
1                             \ real part = 1, imaginary part = 2
4  20                         \ number of directions and lags
  90.0  0.0  4.5              \ azm, dip, lag distance
  45.0  0.0  8.5              \ azm, dip, lag distance
 135.0  0.0  4.5              \ azm, dip, lag distance
   0.0  0.0  4.5              \ azm, dip, lag distance
1  0.0 -0.0025962 0.0018629 0.0 \ nst, nugget effect, c1, c2, c3
3  21.5  45.0  0.0  0.0       \ it, cc, ang1, ang2, ang3
 294.000 105.000 0.0         \ a_hmax, a_hmin, a_vert

```

The output of this program contains alternatively the real component or the imaginary component of the complex covariance model in (4) according to the specification given at the second line of the parameter file, which is set equal to 1 for the real part computation, or equal to 2 for the imaginary part computation. Moreover, apart from some parameters that are similar to the ones used in **GSLib** (Deutsch and Journel 1998), such as the ones regarding the lags and directions where to calculate the components of a complex covariance model, the following details are also given:

- the parameter `nst`, which corresponds to the number of structures of \tilde{C} , used in (4);
- the parameter associated with the nugget effect;
- the parameters `c1`, `c2`, `c3`, which represent the components of the translating vector \mathbf{c} ;
- the parameter `it`, which corresponds to the model type of each structure (Table 1);
- the parameter `cc`, which corresponds to the contribution of each structure;
- the parameters `ang1`, `ang2`, `ang3`, which correspond to the spatial angles, and `a_hmax`, `a_hmin`, `a_vert`, which correspond to the ranges or scales of variability of each structure.

Run ccovamodel

After specifying the parameter file, the user can run the program `ccovamodel` and type the name of the ad-hoc parameter file.

The output is saved in a text file whose extension (usually `.var`) is defined in the parameter file.

In the program `ccovamodel`, some specific error messages may be issued:

- to acknowledge if the indicated model component does not respect the prefixed code;
- to point out if the number of structures is not valid (that is equal or less than zero).

3.5. Program ckriging

The program `ckriging` can perform simple and ordinary complex kriging. According to what has been presented in Section 2, the program `ckriging` allows the system of equations (7)

Type of covariance model	Integer code	
	real part	imaginary part
Spherical	1	5
Exponential	2	6
Gaussian	3	7
Hole effect	4	8

Table 1: Types of covariance models and the corresponding integer codes.

to be used in case of simple complex kriging. Alternatively, in case of ordinary complex kriging, the system of equations (6) can be used. The solutions of the systems (6) and (7) with respect to the parameter vector ω and the Lagrange multiplier M (in case of complex ordinary kriging) require the knowledge of the complex covariance model. The parameters, required by the program `ckriging` for setting the correlation structure given in (4) and the kriging options, are clarified below.

Parameter, input and output files

In the following, the input and output files of the program `ckriging` and the associated parameter file are described.

First of all, the input files are:

- the data file which contains information on the sample locations and the observed values for the components U and V , in **Geo-EAS** format;
- the file for jackknife estimation in **Geo-EAS** format, in case the jackknife option is set. The jackknife file contains information on the locations where the estimation is required and the available true values for the components U and V to be used for comparison.

These files are ASCII files (or text files), their names can be arbitrarily fixed and their extension is usually `.dat`.

The output files are:

- the file which contains some descriptive and inferential statistics for the true data values (if available) and the estimates. Moreover, in case of cross-validation and jackknife, a statistical test on the differences between the means of the true values and the estimated ones for the components U and V is conducted and the p values for the null hypotheses that the differences are equal to zero are computed;
- the debug file, with the usual **GSLib** debugging levels;
- the file which contains the estimated values for the two components U and V and the estimation variance. Moreover, if the kriging option is set to cross-validation or jackknife, the true values for the components U and V and the estimation errors for the two components are also indicated in the output file, as shown in the output example given below.

COMPLEX KRIGING

```

x
y
z
TrueU
TrueV
U Estimate
V Estimate
EstimationVariance
Error: est-trueU
Error: est-trueV
  53.00  39.00  .00  -.4616  1.6783  -.5005  1.3489  1.4229  -.0389  -.3294
  66.00  39.00  .00  -.1640  1.5120  -.1700  1.8259  .0421  -.0060  .3139
  79.00  39.00  .00  .1263  2.2374  .1737  2.6623  .7735  .0474  .4249
  92.00  39.00  .00  .4093  3.2016  .3994  2.4358  .0374  -.0099  -.7658
 105.00  39.00  .00  .6849  2.5404  1.6628  2.9389  .7735  .9779  .3985
 118.00  39.00  .00  .9532  4.0310  1.1270  2.9399  .0374  .1738 -1.0911
 131.00  39.00  .00  1.2142  2.9934  .0848  2.2222  .7735 -1.1294  -.7712
 144.00  39.00  .00  .2958  1.6124  .8884  2.5220  .0374  .5926  .9096
 157.00  39.00  .00  .5421  1.4723  .7554  1.2192  .7735  .2133  -.2531
 170.00  39.00  .00  .2091  .1548  .4185  .1163  .0374  .2094  -.0385
 183.00  39.00  .00  .4407 -1.1309  .8241  -.3324  .7735  .3834  .7985
.....
261.00 338.00 .00 .1865  2.6453  .0842  1.9333  .2879  -.1023  -.7120
274.00 338.00 .00 .6357  .3571  .5030  .0521  .1682  -.1327  -.3050
287.00 338.00 .00 .8955  -.6724  .7825  -.9274  .2198  -.1130  -.2550

```

The parameter file required by the program *ckriging*, in order to provide cross-validation and spatial predictions of a complex random field, is illustrated below.

Parameters for ckriging

START OF PARAMETERS:

```

data10x12.txt          \ file with data
1  2  0  4  5          \ columns for 3D coordinates and U, V
-100.01  1.0e21        \ trimming limits (tmin, tmax)
3                      \ koption: 0 = grid, 1 = cross, 2 = jack_cross, 3 = jack
data19x24.txt          \ file with jackknife data (jackfl)
1  2  0  4  5          \ columns for 3D coordinates and U, V
3                      \ debug level (0, 1, 2, 3): 0 - less => 3 - more info
complex_jack3.dbg      \ debugging output file
complex_jack3.out      \ output file with estimation results
statistics_jack3.out   \ output file with statistics
10  53  26             \ nx, xmn, xsiz
13  39  26             \ ny, ymn, ysiz
1   0   1              \ nz, zmn, zsiz
1   1   1              \ x, y, and z block discretization

```

```

3      8                                \ #_min points, #_max points
200.0 100.0 0.0                        \ maximum search radii
45.0   0.0 0.0                          \ angles for search ellipsoid
1                                           \ kriging type (0 = SK, 1 = OK)
0.00 0.00                               \ mean value(i), i = 1, 2 (nvar = 2)
1                                           \ model for real part (= 1), imag part (= 2)
1 0.0 -0.002596 0.001863 0.0 \ nst, c0, c1, c2, c3
3 21.5 45.0 0.0 0.0 \ it, cc, ang1, ang2, ang3
    294.000 105.000 0.0 \ a_hmax, a_hmin, a_vert

```

Note that some parameters are clearly explained through the brief documentation reported on the right side of the parameter file; on the other hand, some specific parameters, required by the program `ckriging`, are described in order of appearance in the following list:

- the kriging option: `koption = 0` for complex kriging for a grid of points or blocks, `koption = 1` for cross-validation with the data in `datafl`, `koption = 2` or `koption = 3` for jackknife with the data indicated in the jackknife file. If some data locations coincide with some points in the jackknife file, then the option 2 of jackknife allows the coincident data to be removed one at a time and estimated from the remaining neighboring data as in cross-validation, while the option 3 of jackknife allows the coincident data to be included in the searching neighborhood (this option produces exact estimates);
- the name of the output file with the statistics regarding the available data and the estimated data;
- in the subsequent 3 lines, the definition of the grid in an ad-hoc coordinate system, which is necessary for grid predictions (`koption = 0`); this is done in terms of the coordinates at the center of the first node/block (`xmn`, `ymn`, `zmn`), the number of grid nodes/blocks (`nx`, `ny`, `nz`) and the size/spacing of the nodes/block (`xsiz`, `ysiz`, `zsiz`). More details are given in the **GSLib** manual ([Deutsch and Journel 1998](#));
- in the last 6 lines, the kriging type and the complex covariance model:
 - the kriging type is set to 0 if simple kriging is performed; it is set to 1 if ordinary kriging is performed;
 - in case of simple kriging, the known mean values for the two components must be defined;
 - the specifications of the complex covariance model in (4) can be provided for the real part or alternatively for the complex part, since given the translating vector \mathbf{c} , the complex part can be defined starting from the real part and vice versa; hence, the indicator is set to 1 if the specifications are given for the real part, while it is set to 2 if the specifications are given for the imaginary part;
 - the parameter `nst` corresponds to the number of structures of \tilde{C} , used in (4);
 - the parameter `c0` corresponds to the nugget effect;
 - the parameters `c1`, `c2` and `c3` represent the components of the translating vector \mathbf{c} ;
 - the parameter `it` corresponds to the model type of each structure (Table 1);

- the parameter `cc` corresponds to the contribution of each structure of \tilde{C} ;
- the parameters `ang1`, `ang2`, `ang3` correspond to the spatial angles and `a_hmax`, `a_hmin`, `a_vert` correspond to the ranges or scales of variability of each structure of \tilde{C} .

In Table 1, the integer codes that can be used to specify the covariance models for the \tilde{C} in (4) are provided. It is worth highlighting that the admissible complex covariance models, generated through Equation 4, have been implemented in the subroutine `ccov3` which is called by the program `ckriging`. Note that the same subroutine `ccov3` is also called by the program `ccovamodel` described in Section 3.4.

Run `ckriging`

If the parameter file has been suitably filled, then the user can run the program `ckriging` and type the name of the prepared parameter file. The parameters set in the ad-hoc file are read and simultaneously displayed on the screen, then the input data files are read and saved in a temporary array, and some basic statistics are computed, as shown in the following template:

```

Variable          1 in data file:          4
Number            =      120.000000
Average           =      3.023375E-01
Std. deviation    =      1.861279
Std. error        =      1.699108E-01
Minimum value     =      -3.611300
Maximum value     =      5.073600
Variable          2 in data file:          5
Number            =      120.000000
Average           =      -4.459241E-02
Std. deviation    =      4.246270
Std. error        =      3.876297E-01
Minimum value     =     -10.209100
Maximum value     =      7.186400
Variable          1 in jackknife file:      4
Number            =      456.000000
Average           =      1.856544E-01
Std. deviation    =      1.903301
Std. error        =      8.913024E-02
Minimum value     =      -3.630000
Maximum value     =      6.635800
Variable          2 in jackknife file:      5
Number            =      456.000000
Average           =     -9.362296E-03
Std. deviation    =      4.281832
Std. error        =      2.005152E-01
Minimum value     =     -10.209100
Maximum value     =      8.978900

```

As soon as the program starts the kriging procedure, the note `Working on the complex`

kriging is shown; moreover, a report on the working progress from time to time is also provided on the screen, as given below:

```
Working on the complex kriging
  currently on estimate      12
  currently on estimate      24
  currently on estimate      36
  ....
  currently on estimate      456
```

At the end, some descriptive statistics on the estimates of the two components are shown on the screen and written to the debug file, then a closing message on the screen announces that the program has terminated, as shown in the following output template.

DESCRIPTIVE STATISTICS for est. values of U and V

```
Variable U
  Estimated      456 total blocks
  Average        = 2.409850E-01
  Std. deviation = 1.856171
  Std. error     = 8.692319E-02
  Minimum value  = -3.773563
  Maximum value  = 5.201846
```

```
Variable V
  Estimated      456 total blocks
  Average        = 4.879891E-02
  Std. deviation = 4.252596
  Std. error     = 1.991460E-01
  Minimum value  = -10.384210
  Maximum value  = 10.254780
```

TESTING HYPOTHESIS ON DIFFERENCE BETWEEN MEANS

```
P-value (H0: True mean_U = Est mean_U) = .6588
P-value (H0: True mean_V = Est mean_V) = .8716
```

STATISTICS on the errors: True vs Est.

```
MAE for U      = .4186
RMSE for U     = .6595
```

```
MAE for V      = .6260
RMSE for V     = 1.0926
```

CKRIGING Version: 1.000 Finished

Stop - Program terminated.

The output estimates and some statistics regarding available data and estimated data are saved in two separate text files whose extensions (usually `.out` and `.txt` respectively) are defined in the parameter file.

As specified at the beginning, when the program `ckriging` is run, some error messages might appear. In particular, for this specific program, some alerts may be issued

- to point out if the kriging option or the kriging type are not valid;
- to detect if the indicated model type of a basic structure does not exist (it is not one of the models in Table 1).

4. An application on complex data

In the following case study, a vectorial data set with two components is interpreted as a finite realization of a complex-valued random field W , with components U and V (or modulus and direction in a polar system), and is analyzed by using the software `cgeostat`. A directory, called `replication_materials`, contains the executable files, the parameter files, the input and output files used for the case study of this article.

4.1. Data set

The whole data set of the spatial vectorial values (`data19x24.txt`) refers to 456 locations regularly distributed over a grid (19×24). However, a grid of (10×12) is selected and the corresponding vectorial data (`data10x12.txt`) is used for structural analysis; the remaining data is considered for comparison purposes in jackknife prediction.

The vectorial representation of the available complex data over the (19×24) grid is produced by applying the program `clocmap` with the parameter file `clocmap19x24.par` given in Section 3.1. On the other hand, the location map for the selected complex data is similarly obtained by applying the program `clocmap` with the parameter file `clocmap10x12.par` given below.

Parameters for CLOCMAP

START OF PARAMETERS:

```

data10x12.txt          \ file with data
1  2  0  4  5    0    \ columns for coords X, Y, Z & vars U, V, E
-9990.  1.0e21      \ trimming limits for U
-9990.  1.0e21      \ trimming limits for V
-9990.  1.0e21      \ trimming limits for E
0                    \ option overlay: 0 = no, 1 = yes
XXXX.txt             \ file with conditioning data
1  2  0  4  5          \ columns for coords X, Y, Z & vars U, V
1                    \ color code for conditioning data U, V
clocmap10x12.ps      \ file for postscript output
50.0    290.         \ xmn, xmx
35.0    340.         \ ymn, ymx

```

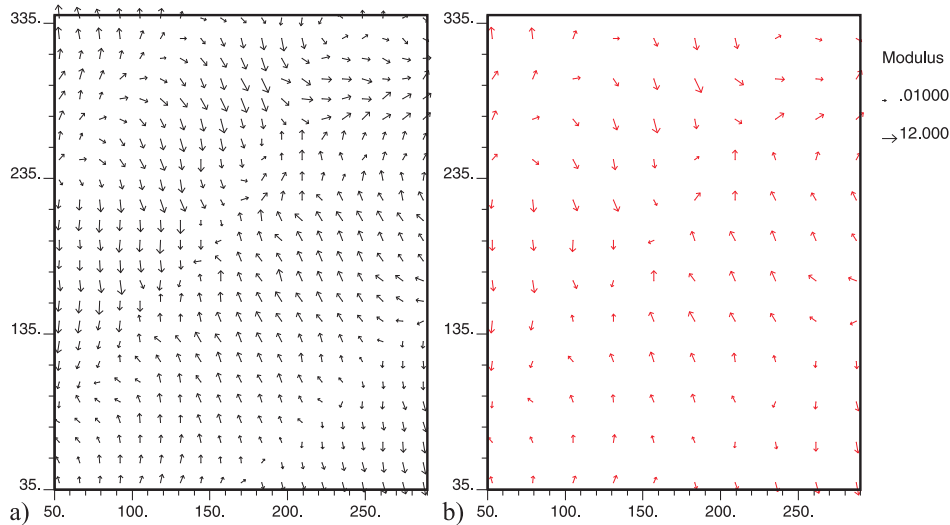


Figure 1: Location map of the available data set over a grid of a) (19×24) , b) (10×12) .

```

0.0          \ z selection
8           \ color code for vector U, V
0           \ 0 = arithmetic, 1 = log10 scaling(|u|>1, |v|>1)
0.01  12.0  3  10 \ modulus scale & size: dmin, dmax, smin, smax
0           \ 0 = no labels, 1 = label each location
8           \ color code for variable E
1           \ transform var.E: 0 = none, 1 = sqrt, 2 = log10
0.5        \ label size of var.E: 0.1 = sml, 1 = reg, 10 = big
Location map of complex data \ title

```

The output files are shown in Figure 1.

In the following, the complex covariance function is estimated and fitted, then complex predictions, based on the selected complex covariance model, are computed over (19×24) grid nodes by using complex kriging implemented in the program `ckriging`.

4.2. Structural analysis and modeling

Given the class of models in (4), a suitable complex covariance function for the data under study is found through

- the estimation of the covariance functions for components U and V of the complex random field W , denoted as C_U and C_V , and the corresponding cross-covariance functions C_{UV} and C_{VU} for different lags and directions of the spatial domain;
- the estimation of the components, C^{Re} and C^{Im} , of the complex covariance function given in (2), by computing the sum of the sample direct covariances \hat{C}_U and \hat{C}_V and the difference of the sample cross-covariances \hat{C}_{VU} and \hat{C}_{UV} for relevant directions, and fitting of the parameter vector \mathbf{c} ;
- selection of an appropriate anisotropic covariance model $\tilde{C}(\cdot; \boldsymbol{\theta})$ with parameter vector

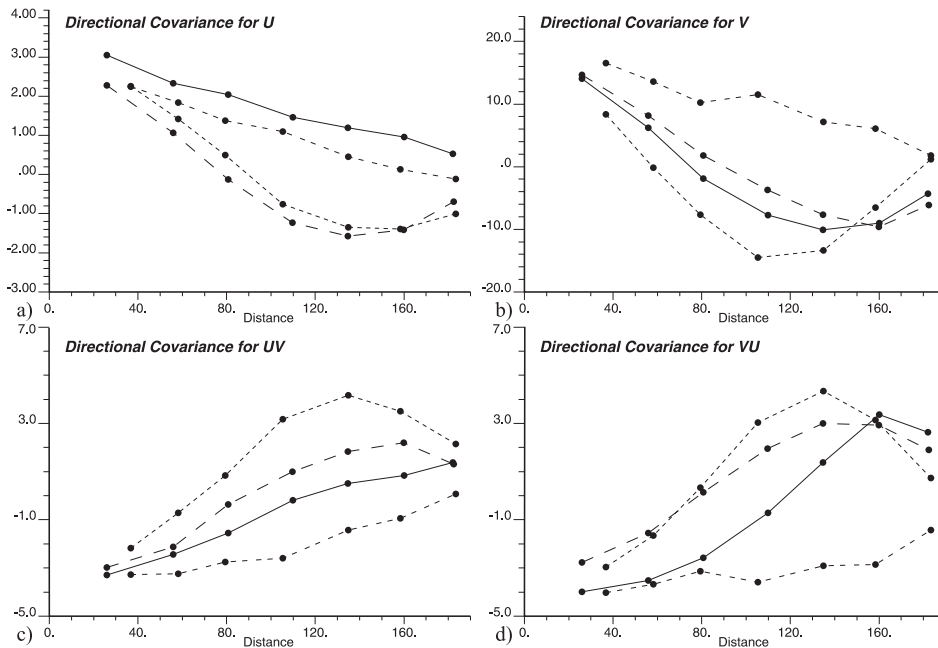


Figure 2: Sample covariance functions of a) U and b) V , and cross-covariance functions of c) UV and d) VU for different directions: _____ East, _____ North-East, _____ South-East, _____ North.

θ , by inspecting the directional sample covariance functions, and fitting it to the sum of the sample direct covariances and the difference of the sample cross-covariances.

The sample covariance functions \hat{C}_U and \hat{C}_V and the corresponding cross-covariance functions \hat{C}_{UV} and \hat{C}_{VU} for different lags and directions are estimated by using the original **GSLib** program `gamv` with the appropriate parameter files. In particular, four parameter files are filled in for directional sample covariance functions of the components U (`covaU.par`) and V (`covaV.par`), as well as for directional sample cross-covariance functions of UV (`crosscovaUV.par`) and VU (`crosscovaVU.par`).

The graphical representations of the estimated direct and cross covariance functions are obtained by applying the program `ccovaplt` with ad-hoc parameter files (Figure 2). Hence, four parameter files are filled in for plotting the directional sample covariance functions of the components U (`plt_covaU.par`) and V (`plt_covaV.par`), and the directional sample cross-covariance functions of UV (`plt_covaUV.par`) and VU (`plt_covaVU.par`). For example the parameter file `plt_covaU.par`, which is used to plot the directional sample covariance function of the component U , is given below.

Parameters for ccovaplt

START OF PARAMETERS:

```
plt_covaU.ps      \ file for postscript output
4                \ number of covariances to plot
0.0  190.0       \ distance limits (from data if max < min)
```

```

-3.0    5.0          \ covariance limits (from data if max < min)
Directional Cov for U \ title for covariance function
1          \ comp. on data to plot: none = 1, sum = 2, dif = 3
covaU.txt  \ 1 file with covariance data
XXX.txt    \ 1 file with covariance data (only for type = 2, 3)
1    0    1    1    8 \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
1          \ comp. on data to plot: none = 1, sum = 2, dif = 3
covaU.txt  \ 1 file with covariance data
XXX.txt    \ 1 file with covariance data (only for type = 2, 3)
2    0    1    1    1 \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
1          \ comp. on data to plot: none = 1, sum = 2, dif = 3
covaU.txt  \ 1 file with covariance data
XXX.txt    \ 1 file with covariance data (only for type = 2, 3)
3    0    1    1    2 \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr
1          \ comp. on data to plot: none = 1, sum = 2, dif = 3
covaU.txt  \ 1 file with covariance data
XXX.txt    \ 1 file with covariance data (only for type = 2, 3)
4    0    1    1    3 \ covariance#, dash#(1-10), pts(1-0), line(1-0), colr

```

In this case, the input files for the program `ccovaplt` correspond to the output files of the program `gamv`. The reader can find the other parameter files in the supplementary material in directory `replication_materials` together with the case study results.

Then the real and imaginary components of the complex covariance are estimated and modeled. In particular, given the class (4) of complex covariance models, the least squares technique is first used for c_1 and c_2 estimation. The program `cfitting` with the parameter file given in Section 3.2, provides the least squares estimates for c_1 and c_2 . Then the following complex covariance model is fitted:

$$C(\mathbf{h}; \mathbf{c}, \boldsymbol{\theta}) = \cos(\mathbf{h} \cdot \mathbf{c})\tilde{C}(\mathbf{h}; \boldsymbol{\theta}) + i \sin(\mathbf{h} \cdot \mathbf{c})\tilde{C}(\mathbf{h}; \boldsymbol{\theta}), \quad (8)$$

where $\tilde{C}(\mathbf{h})$, with $\tilde{C}(\mathbf{0}) = 21.5$, is a Gaussian covariance model with geometric anisotropy, whose direction of maximum continuity is 45° with maximum range equal to 98 (corresponding to an effective range of 294) and minimum range in the perpendicular direction equal to 35 (corresponding to an effective range of 105), that is $\boldsymbol{\theta} = (21.5, 294, 105)$, $\mathbf{c} = (-0.0025962, 0.0018629, 0.0)$. Note that the geometric characteristics of $\tilde{C}(\mathbf{h})$ are fixed through visual inspection of the empirical direct and cross covariance functions. In particular, the real and imaginary parts of the complex covariance model are given below:

$$\begin{aligned}
C^{Re}(h_x, h_y; 45^\circ) &= 21.5 \cos(-0.0025962h_x + 0.0018629h_y) \exp[-(h/98)^2], \\
C^{Re}(h_x, h_y; 135^\circ) &= 21.5 \cos(-0.0025962h_x + 0.0018629h_y) \exp(-(h/35)^2), \\
C^{Im}(h_x, h_y; 45^\circ) &= 21.5 \sin(-0.0025962h_x + 0.0018629h_y) \exp(-(h/98)^2), \\
C^{Im}(h_x, h_y; 135^\circ) &= 21.5 \sin(-0.0025962h_x + 0.0018629h_y) \exp(-(h/35)^2),
\end{aligned}$$

where $h = \sqrt{h_x^2 + h_y^2}$.

The program `ccovamodel` is applied to compute the real and imaginary parts of the complex covariance model. The parameter file used to produce the real component is given in Section 3.4, while the parameter file used to determine the imaginary part is illustrated below.

Parameters for ccovamodel

START OF PARAMETERS:

```

ccovamodel_im.var      \ file for covariance output
2                      \ real part = 1, imaginary part = 2
4  20                  \ number of directions and lags
  90.0  0.0  4.5      \ azm, dip, lag distance
  45.0  0.0  8.5      \ azm, dip, lag distance
 135.0  0.0  4.5      \ azm, dip, lag distance
   0.0  0.0  4.5      \ azm, dip, lag distance
1  0.0 -0.0025962 0.0018629 0.0 \ nst, nugget effect, c1, c2, c3
7  21.5  45.0  0.0  0.0      \ it, cc, ang1, ang2, ang3
  294.000  105.000  0.0      \ a_hmax, a_hmin, a_vert

```

Both parameter files are filled in for computing the covariance model along 4 directions and 20 lags. Note that the model specifications fixed in these parameter files are also used for complex kriging (see Section 4.3). As specified before, the output of this program contains alternatively the real component or the imaginary part of the complex covariance model (4) in accordance with the specification given at the second line of the parameter file, which is set equal to 1 for the real part computation, or equal to 2 for the imaginary part computation.

The graphical representations of the sample directional components of the complex covariance function and the corresponding model are obtained by using the program *ccovaplt*. Four parameter files are appropriately filled in, one for each direction. For example, the parameter file *plt_ccova_SE.par* is illustrated in Section 3.2 and it is prepared so that the sample directional components (real and imaginary) of the complex covariance function and the corresponding model along the direction South-East are plotted. Note that, for a fixed direction, the sample real component is obtained by summing, lag by lag, the sample direct covariance functions for U and V , while the sample imaginary component is obtained by applying the difference, lag by lag, of the sample cross-covariance function for VU with respect to the one for UV . This reflects the theoretical equations given in (3). The reader can find the other three parameter files in the attached directory with the case study results. Figure 3 shows the real and imaginary parts of the sample complex covariance function and the corresponding models computed for different directions.

In Section 4.3, cross-validation, grid predictions and jackknife predictions are computed by applying the above model characterizations.

4.3. Complex validation and prediction

The program *ckriging* is used for both cross-validation and prediction purposes (by using the grid option or the jackknife options) with the parameter files appropriately filled in.

First of all, the complex covariance model for the corresponding random field is cross-validated on the basis of the available data at the 120 spatial points located over a grid (10×12). The parameter file *ckriging_cross.par* used for cross-validation is presented below.

Parameters for ckriging

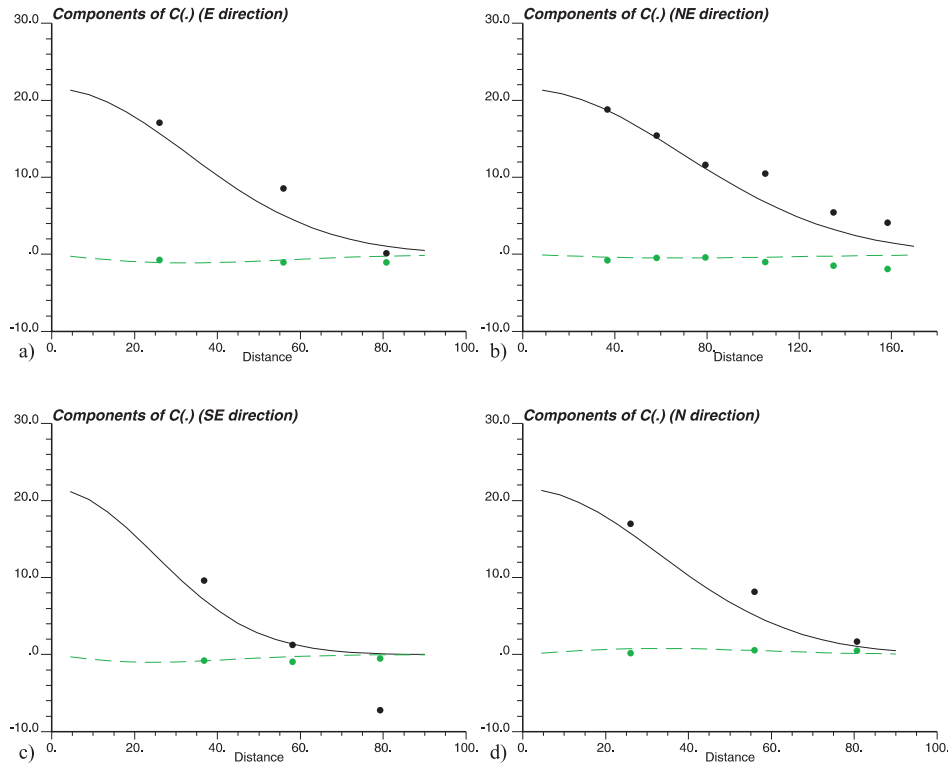


Figure 3: Real part of the sample complex covariance function (\bullet) and its model (____) with the corresponding imaginary part (\bullet) and its model (____) for different directions: a) East, b) North-East, c) South-East, d) North.

START OF PARAMETERS:

```

data10x12.txt          \ file with data (datafl)
 1  2  0  4  5          \ columns for 3D coordinates + 2 components
-100.01    1.0e21      \ trimming limits (tmin, tmax)
1                  \ koptio: 0 = grid, 1 = cross, 2 = jack_cross, 3 = jack
XXX.txt             \ file with jackknife data (jackfl)
0  0  0  0  0        \ columns for 3D coordinates + 2 components
3                  \ debug level (0, 1, 2, 3)
complex_cross.dbg   \ debugging output file
complex_cross.out   \ output file
statistics_cross.out \ file for statistics
 10   53   26        \ nx, xmn, xsiz
 13   39   26        \ ny, ymn, ysiz
  1    0    1        \ nz, zmn, zsiz
1    1    1          \ x, y, and z block discretization
3    8              \ #_min points, #_max points
200.0 100.0 0.0     \ maximum search radii
 45.0  0.0 0.0     \ angles for search ellipsoid
1                  \ kriging type (0 = SK, 1 = OK)

```

```

0.00  0.00          \ mean value(i), i = 1, 2 (nvar = 2)
1          \ model for real part (= 1), imag part (= 2)
1 0.0 -0.0025962 0.001862 0.0 \ nst, c0, c1, c2, c3
3 21.5  45.0  0.0  0.0 \ it, cc, ang1, ang2, ang3
  294.00  105.00  0.0  \ a_hmax, a_hmin, a_vert

```

Note that

- the complex kriging option is set equal to 1, in order to produce cross-validation;
- the parameters related to jackknife predictions (such as the file name and the columns for the coordinates and the components U and V in the jackknife file) do not have to be specified in this case;
- the file names which contain the debugging data, the cross-validation results and the corresponding statistics are set equal to `complex_cross.dbg`, `complex_cross.out` and `statistics_cross.out`, respectively;
- regarding the grid specifications, they are useful in **GSLib** algorithms for defining local search neighborhoods apart from the selected kriging option. In this case, the coordinates at the first point are (53, 39, 0), the grid nodes are 10 along the East direction, 13 along the North direction and 0 in the vertical direction (indicating that a 2D grid is defined) and the spacing between the nodes are, respectively, 26, 26 and 1. The size of this search network is usually much larger than the final estimation or simulation grid node spacing (Deutsch and Journel 1998);
- regarding block discretization, the value 1 results in point kriging being performed;
- the characteristics of the search neighborhood are such that the estimation is performed by using from 3 to 8 sample data which fall in the ellipsoid whose maximum axis, along the direction of maximum continuity (corresponding to 45° clockwise from the azimuth direction) is 200, and the minimum axis is 100. These values are fixed taking into account the ranges of the covariance model;
- regarding the complex covariance model, the parameters are such that the fitted model (8) is considered.

Tables 2 and 3 illustrate some descriptive and inferential statistics for the original data values and the estimates, which are computed and saved in the output file `statistics_cross.out`. They can be compared and used to assess the estimation procedure and indirectly the fitted covariance function.

Indeed, a statistical test is conducted in order to verify that the differences between the means of the true values and the estimated ones for the components U and V are nil. The p values of the test-statistics are determined (Tables 2 and 3) indicating that the null hypothesis is retained. Hence, the complex covariance function can be considered appropriate for modeling the spatial correlation of the analyzed vectorial components, as also confirmed by computing the mean absolute error (MAE) and the root mean square error (RMSE) between the true values and the estimated ones for the two components (Tables 2 and 3).

Cross-validation statistics	Component X		Component Y	
	True	Est.	True	Est.
Num. of data	120	120	120	120
Mean	0.302	0.251	-0.046	0.007
Std. dev.	1.861	1.967	4.264	4.139
Std. err.	0.170	0.180	0.388	0.378
Minimum	-3.611	-4.020	-10.209	-9.689
Maximum	5.074	5.976	7.186	6.884
<i>Mean difference test</i>				
Hypoth. difference		0		0
<i>p</i> value		0.8361		0.9245
MAE		1.0228		0.9781
RMSE		1.4027		1.3309

Table 2: Some statistics on cross-validation estimations.

Jackknife statistics	Component X			Component Y		
	True	Est. (op. 2)	Est. (op. 3)	True	Est. (op. 2)	Est. (op. 3)
Num. of data	456	456	456	456	456	456
Mean	0.212	0.236	0.250	0.045	0.055	0.042
Std. dev.	1.908	1.878	1.850	4.262	4.231	4.259
Std. err.	0.089	0.088	0.087	0.200	0.198	0.199
Minimum	-3.630	-4.020	-3.611	-10.209	-10.448	-10.448
Maximum	6.636	5.976	5.246	8.979	9.200	9.200
<i>Mean difference test</i>						
Hypoth. difference		0	0		0	0
<i>p</i> value		0.8465	0.7615		0.9709	0.9908
MAE		0.6525	0.3834		0.8585	0.6011
RMSE		0.9422	0.6082		1.2578	1.0563

Table 3: Some statistics on jackknife estimations with the kriging options 2 and 3.

After cross-validation, the new program `ckriging` is applied to predict the variable under study through complex ordinary kriging.

First of all, the grid option is considered; in particular, the above discussed covariance model is used for the estimation of the two components U and V over a grid (19×24). The parameter file used for the grid prediction is given below.

Parameters for ckriging

START OF PARAMETERS:

```

data10x12.txt          \ file with data (dataf1)
1  2  0  4  5          \ columns for 3D coordinates + 2 components
-100.01    1.0e21      \ trimming limits (tmin, tmax)
0                                                    \ koption: 0 = grid, 1 = cross, 2 = jack_cross, 3 = jack
    
```

```

XXX.txt                \ file with jackknife data (jackfl)
0 0 0 0 0             \ columns for 3D coordinates + 2 components
3                     \ debug level (0, 1, 2, 3)
complex_grid.dbg      \ debugging output file
complex_grid.out      \ output file
statistics_grid.out   \ file for statistics
 19 53 13             \ nx, xmn, xsiz
 24 39 13             \ ny, ymn, ysiz
 1 0 1                \ nz, zmn, zsiz
1 1 1                \ x, y, and z block discretization
3 8                   \ #_min points, #_max points
200.0 100.0 0.0      \ maximum search radii
 45.0 0.0 0.0        \ angles for search ellipsoid
1                     \ kriging type (0 = SK, 1 = OK)
0.00 0.00            \ mean value(i), i = 1, 2 (nvar = 2)
1                     \ model for real part (= 1), imag part (= 2)
1 0.0 -.0025962 0.001862 0.0 \ nst, c0, c1, c2, c3
3 21.5 45.0 0.0 0.0 \ it, cc, ang1, ang2, ang3
 294.00 105.00 0.0 \ a_hmax, a_hmin, a_vert

```

Successively, the entire data set (not used for structural analysis) is also considered for the jackknife options 2 and 3. The two parameter files used for jackknife predictions, which are named `ckriging_jack2.par` and `ckriging_jack3.par`, are equal to the previous one except for the kriging option which is set equal to 2 for the former and 3 for the latter (the parameter file is illustrated in Section 3.5).

In Table 3, some descriptive and inferential statistics regarding the jackknife results, with both options, are given. It is worth underlining that the estimation results over the grid (19×24) obtained through the grid option (`koption = 0`) are equal to the ones obtained with the jackknife option 3 (`koption = 3`) over the same grid (19×24), except that this last option provides the comparisons with the true values.

The statistical test on the mean difference of the two components confirms that the null hypothesis of zero means can be accepted with low margin of uncertainty, in terms of p values. Moreover, note that both MAE and RMSE highlight the capability of the algorithm to reproduce the vectorial random field. Thus, the jackknife results show that the method can recover the unknown truth when every part of the model is correctly specified.

Figure 4 shows the prediction map of the complex-valued random field for the study area, together with the error standard deviation (square root of the error variance), and the conditioning data (available data over the grid (10 × 12)), marked with red arrows. This location map is obtained by using the program `clocmap` with the parameter file `clocmap_jack.par` given below.

Parameters for CLOCMAP

START OF PARAMETERS:

```

complex_jack3.out     \ file with data
1 2 0 6 7 8          \ columns for coords X, Y, Z & vars U, V, E
-9990. 1.0e21         \ trimming limits for U

```

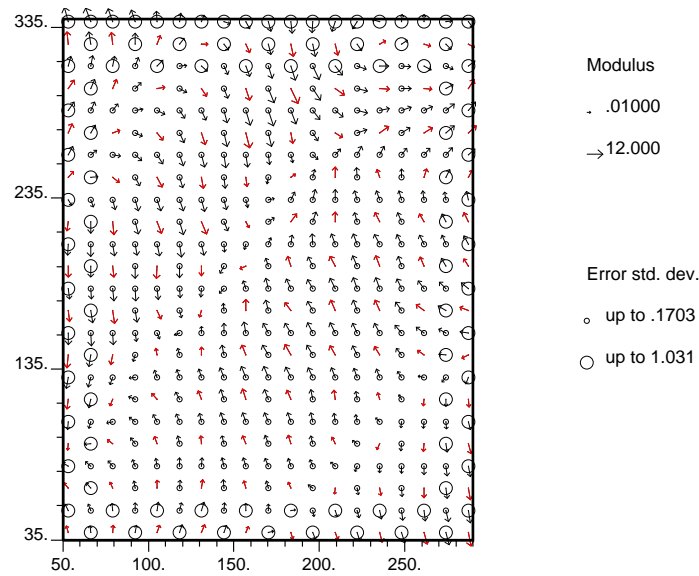


Figure 4: Location map of the predicted values of the complex random field W , through complex kriging (\rightarrow), and the conditioning data over a grid (10×12) (\rightarrow).

```

-9990.    1.0e21          \ trimming limits for V
-9990.    1.0e21          \ trimming limits for E
0          \ option overlay: 0 = no, 1 = yes
data10x12.txt          \ file with conditioning data
1  2  0  4  5          \ columns for coords X, Y, Z & vars U, V
1          \ color code for conditioning data U, V
clocmap_jack.ps        \ file for postscript output
50.0    290.          \ xmn, xmx
35.0    340.          \ ymn, ymx
0.0          \ z selection
8          \ color code for vector U, V
0          \ 0 = arithmetic, 1 = log10 scaling(|u|>1, |v|>1)
0.01    12.0  3  10   \ modulus scale & size: dmin, dmax, smin, smax
0          \ 0 = no labels, 1 = label each location
8          \ color code for variable E
1          \ transform var.E: 0 = none, 1 = sqrt, 2 = log10
0.5          \ label size of var.E: 0.1 = sml, 1 = reg, 10 = big
          \ title

```

5. Summary

In this paper, some dedicated Fortran programs for geostatistical analysis of a finite realization of a complex-valued random field are presented. Complex kriging and some other programs, for representing complex data, computing and plotting components of a sample complex covariance function, computing and plotting components of a complex covariance model,

have been implemented and an application on vectorial data with two components has been proposed. In particular, these routines are customized programs generated, on the basis of the **GSLib** library (specifically the version **GSLIB77**), for analyzing complex-valued random variables on a spatial domain \mathbb{R}^d ($d \leq 3$).

The new **GSLib** programs `ckriging`, `clocmap`, `ccovaplt`, `cfitting` and `ccovamodel`, provide some computational solutions, which had not been implemented in the **GSLib** library before. In particular, the program `ckriging` faces the problem of estimating a complex-valued random field. Differently from the original kriging program `KT3D`, the following computational aspects have been considered in the program `ckriging`:

- complex kriging systems, given in (6) and (7), have been implemented;
- the program `COVA3`, which belongs to the **GSLib** library, has been modified in order to include the admissible complex covariance model, defined in expression (4). This modified program has been renamed `ccov3` and it has been included in the main program `ckriging.for`. In this way, it has not been necessary to modify the **GSLib** library;
- simultaneous estimations of the two variables, which are the two components of the complex-valued random field, and the associated error variance are computed;
- another kriging option for jackknife estimation has been included: the option 3 of jackknife allows the coincident data (the ones which coincides with the data locations) to be included in the searching neighborhood. This produces exact estimates associated with the jackknife points which are coincident with that data locations;
- at the end, some descriptive statistics on the estimates of the two components are shown on the screen and written in a text file; moreover the mean absolute errors and the mean square errors for both components are computed, shown on the screen and written on the same text file.

Similarly, the other programs provide innovative tools for analyzing and representing complex variables and their correlation structure.

In the future, various complex covariance models and different forms of predictors for complex variables can be considered. It is worth highlighting that the implementation in the R environment and the extension to a spatio-temporal context might be also interesting as well as the idea of understanding if the complex-valued random fields models can adhere to solutions of particular partial differential equations should be investigated in the future.

Acknowledgments

I would like to really thank the editors and the reviewers for the interest demonstrated and the precious suggestions which have contributed to improve the paper.

References

- Austenfeld M, Beyschlag W (2012). “A Graphical User Interface for R in a Rich Client Platform for Ecological Modeling.” *Journal of Statistical Software*, **49**(4), 1–19. doi: [10.18637/jss.v049.i04](https://doi.org/10.18637/jss.v049.i04).

- Barry R (1996). “A Diagnostic to Assess the Fit of a Variogram Model to Spatial Data.” *Journal of Statistical Software*, **1**(1), 1–11. doi:10.18637/jss.v001.i01.
- Chilés JP, Delfiner P (1999). *Geostatistics: Modeling Spatial Uncertainty*. John Wiley & Sons, New York.
- Cressie N, Huang HC (1999). “Classes of Nonseparable, Spatio-Temporal Stationary Covariance Functions.” *Journal of the American Statistical Association*, **94**(448), 1330–1340. doi:10.2307/2669946.
- De Cesare L, Myers DE, Posa D (2002). “Fortran Programs for Space-Time Modeling.” *Computers & Geosciences*, **28**(2), 205–212. doi:10.1016/s0098-3004(01)00040-1.
- De Iaco S, Maggio M, Palma M, Posa D (2012). “Advances in Spatio-Temporal Modeling and Prediction for Environmental Risk Assessment.” In B Haryanto (ed.), *Air Pollution: A Comprehensive Perspective*, chapter 14, pp. 365–390. InTech. doi:10.5772/51227.
- De Iaco S, Myers DE, Palma M, Posa D (2010). “Fortran Programs for Space-Time Multivariate Modeling and Prediction.” *Computers & Geosciences*, **36**(5), 636–646. doi:10.1016/j.cageo.2009.10.004.
- De Iaco S, Palma M, Posa D (2003). “Covariance Functions and Models for Complex-Valued Random Fields.” *Stochastic Environmental Research and Risk Assessment*, **17**(3), 145–156. doi:10.1007/s00477-003-0129-5.
- De Iaco S, Posa D (2012). “Predicting Spatio-Temporal Random Fields: Some Computational Aspects.” *Computers & Geosciences*, **41**(4), 12–24. doi:10.1016/j.cageo.2011.11.014.
- De Iaco S, Posa D, Palma M (2013). “Complex-Valued Random Fields for Vectorial Data: Estimating and Modeling Aspects.” *Mathematical Geosciences*, **45**(5), 557–573. doi:10.1007/s11004-013-9468-z.
- Deutsch CV, Journel AG (1998). **GSLib: Geostatistical Software Library and User’s Guide**. Applied Geostatistics Series, 2nd edition. Oxford University Press, New York.
- Diggle PJ, Ribeiro Jr PJ (2007). *Model Based Geostatistics*. Springer-Verlag, New York.
- Dutter R (1996). *Analysis of Spatial Data Using GEOSAN: A Program System for Geostatistical Analysis*. University of Technology, Vienna, Austria. Handbook.
- Englund E (1991). **Geo-EAS 1.2.1: Geostatistical Environmental Assessment Software. User’s Guide**. Environmental Monitoring Systems Laboratory, Office of Research and Development, U.S. Environmental Protection Agency, Las Vegas, Nevada.
- Gabriel E, Rowlingson B, Diggle PJ (2013). “stpp: An R Package for Plotting, Simulating and Analysing Spatio-Temporal Point Patterns.” *Journal of Statistical Software*, **53**(2), 1–29. doi:10.18637/jss.v053.i02.
- Geovariances (2015). **ISATIS Software**. Geovariances, Avon, France.
- Haslett J, Raftery AE (1989). “Space-Time Modelling with Long-Memory Dependence: Assessing Ireland’s Wind-Power Resource.” *Applied Statistics*, **38**(1), 1–50. doi:10.2307/2347679.

- Lajaunie C, Béjaoui R (1991). *Sur le krigeage des fonctions complexes*. Note N-23/91/G. Centre de Geostatistique, Ecole des Mines de Paris, Fontainebleau.
- Laurent T, Ruiz-Gazen A, Thomas-Agnan C (2012). “**GeoXp**: An R Package for Exploratory Spatial Data Analysis.” *Journal of Statistical Software*, **47**(2), 1–23. doi:10.18637/jss.v047.i02.
- Melo C, Santacruz A, Melo O (2015). **geospt**: *An R Package for Spatial Statistics*. R package version 1.0-2, URL <https://CRAN.R-project.org/package=geospt>.
- Pebesma EJ (2004). “Multivariable Geostatistics in S: The **gstat** Package.” *Computers & Geosciences*, **30**(7), 683–691. doi:10.1016/j.cageo.2004.03.012.
- Pebesma EJ (2012). “**spacetime**: Spatio-Temporal Data in R.” *Journal of Statistical Software*, **51**(7), 1–30. doi:10.18637/jss.v051.i07.
- Pebesma EJ, Bivand RS (2005). “Classes and Methods for Spatial Data in R.” *R News*, **5**(2), 9–13.
- Pebesma EJ, Wesseling CG (1998). “**gstat**: A Program for Geostatistical Modelling, Prediction and Simulation.” *Computers & Geosciences*, **24**(1), 17–31. doi:10.1016/S0098-3004(97)00082-4.
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Remy N, Boucher A, Wu J (2009). *Applied Geostatistics with SGeMS: A User’s Guide*. Cambridge University Press, New York.
- Renard D, Desassis N, Beucher H, Ors F, Laporte F (2014). “**RGeostats**: The Geostatistical Package.” Mines Paris Tech.
- Ribeiro Jr PJ, Diggle PJ (2001). “**geoR**: A Package for Geostatistical Analysis.” *R News*, **1**(2), 15–18.
- Schlather M, Malinowski A, Menck PJ, Oesting M, Strokorb K (2015). “Analysis, Simulation and Prediction of Multivariate Random Fields with Package **RandomFields**.” *Journal of Statistical Software*, **63**(8), 1–25. doi:10.18637/jss.v063.i08.
- Staios LLC (2001). **WinGSLIB**: *Geostatistical Software Library, version 1.4 for Windows*.
- Wackernagel H (2003). *Multivariate Geostatistics: An Introduction with Applications*. Springer-Verlag, Berlin.

A. Data files and parameter files

This appendix provides some **GSLib** standards in terms of the format of the data files and the parameter files required by the executable files.

There is no choice for the user to specify explicitly the input format (usually **Geo-EAS** format), however the accessibility of the source code allows this to be changed. Missing data are codified with large negative or positive numbers (e.g., -9999 , or 9999 , or $-1.0e21$ or $1.0e21$). It is important to highlight that the programs read numerical values and not alphanumeric strings. In the presence of alphanumeric variables, these can be transformed to integers or alternatively the source code can be modified.

All the variables, options and names of input/output files are contained in a parameter file. In the parameter files, the parameter values can be appropriately stated on the left side, while a brief documentation of the parameters are reported on the right side. The user can add as many lines of comments at the top of the parameter file as desired, but formatted input starts immediately after the string “**START**”. Example of parameter files are included in the paper. Each program requires its own parameter file, whose name is given by the user from a keyboard entry. Indeed, consistently to the **GSLib** standards, when one of the provided programs of the **cgeostat** software is run, the following question appears:

```
Which parameter file do you want to use?
```

Then, the user has to type the name of the parameter file, previously filled, with its absolute path or relative to the current working directory. If no file name is typed, then the program would look for the parameter file which has the same name as the program and the extension **.par** (for example, the program **ckriging** would automatically look for **ckriging.par**). If the parameter file is not found a blank parameter file is created and the program stops. Otherwise if the parameter file is found, the parameter values are read from the parameter file and are simultaneously displayed on the screen.

A closing message on the screen announces that the program has terminated, as shown in the following output template

```
Stop - Program terminated.
```

The input and output files are saved in the folder the users are working in, unless a different path is specified in the parameter file.

When a serious error is encountered, an error alert is displayed on the screen and the program is stopped. Sometimes less serious problems or inconsistencies cause a warning to be displayed on the screen or in a debugging file, and the program will continue. In particular, when a program is run, some error messages might appear in order, for example, to acknowledge if the input files are missing or to indicate if the parameter file does not respect the prefixed standard.

However, it is possible to make a mistake that has not been foreseen. In that case, check the input and compare it to the data. If you cannot solve the problem, send the parameter file and data file to sandra.deiaco@unisalento.it.

B. Dependencies of the programs on the rest of the GSLib

As already specified, the new **GSLib** routines call some programs which belong to the **GSLib** library. In particular, the programs in the library which are called by the new program, are linked to the main program through the file `gslib.lib`, located in the subdirectory `gslib` under the directory `DOS` (for Windows binary users), or through the file `libgs.a`, located in the subdirectory `gslib` under the directories `LIN` and `Mac` (for Mac or Linux users).

Moreover, in a preliminary step of structural analysis of a complex random field, direct and cross-covariance functions have to be estimated. As specified in the case study, this can be done by using the original **GSLib** routine `gamv`.

Affiliation:

Sandra De Iaco

Department of Management, Economics, Mathematics and Statistics

Università del Salento

73100 Lecce, Italy

E-mail: sandra.deiaco@unisalento.it

URL: http://www.economia.unisalento.it/personale_docente