G0014

# DOCUMENTATION OF THE MIXED SUPPORT KRIGING PROGRAMS

Christian Lajaunie

October 25, 1996

## Introduction

These programs are designed to deal with data at a limited number of different supports. They are not appropriate if the support of the data are all different one from another. Only one block support will be considered.

It is assumed also that the structural analysis of the data at various supports had been conducted, and that a point variogram model is known.

This document begins by outlining the methodological options retained when writing the software. Then a brief description of the main programs follows, and an example of application is given. The text ends with an appendix giving some information on how to define a nugget effect at the point support, which is always a source of confusion.

## 1 Methodological options.

There is in principle no problem to deal with data on different supports in a linear kriging. The kriging system requires only the calculation of the sample-sample grade covariances :

$$\mathrm{Cov}\{Z(v_i), Z(v_j)\} = \overline{C}(v_i, v_j) = \frac{1}{v_i}\frac{1}{v_j}\int_{v_i}\int_{v_j} C(x-y)\,dxdy$$

and of the sample-block covariances. The drawback is that these calculations require tedious numerical integration. For that reason random kriging had been preferred to classical kriging. This model start with the specification of a block grid. This grid is defined by its origin, the block size and the number of blocks along each axis. Each sample is located in one of these blocks. The model, which is quite similarly to the discretized models used in disjunctive kriging, assumes that the positions of the samples are random uniform and independent within the block in which they are located. This considerably simplifies the covariance calculations. For instance if $\underline{v}_i$ and $\underline{v}_j$ are two randomized samples within the blocks $V$ and $V'$ :

$$\mathrm{Cov}\{Z(\underline{v}_i), Z(\underline{v}_j)\} = \overline{C}(V, V') = \frac{1}{V}\frac{1}{V'}\int_{V}\int_{V'} C(x-y)\,dxdy$$

This formula remains valid if $V = V'$ and $i \neq j$. Note that the sample volumes $|v_i|$ and $|v_j|$ need not be equal. In the case $i = j$ we have :

$$\mathrm{Var}\{Z(\underline{v}_i)\} = \sigma_{v_i}^2 = \frac{1}{|v_i|^2}\int\int_{v_i} C(x-y)\,dxdy$$

We see that only the block covariances and the sample variances are needed. These can be calculated on the block grid up to the maximum distance involved in the kriging neighborhood before the kriging step. The price for this simplification is a very slight loss of optimality.

## 1.1 The mixed support random kriging equations, and the kriging variance.

To write up the equations of the random mixed support kriging, it is convenient to introduce some notation.

- The block to be kriged will be referred to as $V$, while the blocks from the grid containing data which are located inside the kriging neighborhood will be called $V_k, k = 1..., N$. Very often in practice $V$ is in fact one of the $V_k$. But the notation which makes the distinction according to the role of $V$, namely as block to be estimated or as block containing data, will be used, since no confusion can result from this.

- The different sample volumes will be $v_i, i = 1, ..., m$. Each of them has a variance $\sigma_i^2 = \mathrm{Var}\{Z(v_i)\}$. For the block $V_k$, the number of samples with this support will be $n_k^i \geq 0$.

- The samples within $V_k$, of support $v_i$ will be denoted by $Z(v_i^\alpha); \alpha = 1, ..., n_k^i$. All of them will receive the weight $\lambda_k^i$, so that the block estimator is:

$$Z_V^* = \sum_k \sum_i \lambda_k^i \sum_{\alpha=1}^{n_k^i} Z(v_i^\alpha)$$

Then for each block $V_k$ and for each sample support such that $n_k^i > 0$ there is a kriging equation. We have to make the distinction between the following cases :

1. Two different blocks (the covariance is then $\overline{C}(V_k, V_{k'})$).
2. Two samples with different support within $V_k$ (the covariance is the block variance $\sigma_V^2$).
3. Two different samples with the same support within $V_k$ (the covariance is then again $\sigma_V^2$).
4. The diagonal term, $\sigma_i^2$.

Grouping the equal weights gives the following kriging equation :

$$(n_k^i \sigma_i^2 + n_k^i(n_k^i - 1)\sigma_V^2)\,\lambda_k^i + \sum_{i' \neq i} n_k^i n_k^{i'} \sigma_V^2 \lambda_k^{i'} + \sum_{k' \neq k} \sum_{i'} n_k^i n_{k'}^{i'} \overline{C}(V_k, V_{k'})\,\lambda_{k'}^{i'} + n_k^i\,\mu = n_k^i\,\overline{C}(V_k, V)$$

It would be unwise to simplify this equation by dividing by $n_k^i$ since this would destroy the symmetry of the kriging matrix. The universality condition is :

$$\sum_k \sum_i n_k^i\,\lambda_k^i = 1$$

and the kriging variance is given by :

$$\mathrm{Var}\{Z_V - Z_V^*\} = \sigma_V^2 - \sum_{k,i} (\overline{C}(V_k, V) - \mu)\, n_k^i\, \lambda_k^i$$

# 2 The programs.

The programs and the data files have structures very similar to the ones in the discrete disjunctive programs. But because there are some incompatibilities in the formats, the set of programs for mixed support kriging has to be considered as independent.

The kriging is in done in three steps. The first one is the preparation of the data. and the setup of the block structure. The second one is the covariance calculation. The third one is the kriging itself.

## 2.1 Preparing the data files.

First the data must be loaded into unformatted files. One file is necessary for each data support. Two methods can ensure the necessary conversion:

1. The general method is to code a call to the subroutine `wrdata()` (one call per support). This subroutine opens a new file and store the information. The arguments are as follows:

   ```
   wrdata(grades,coordinates,nd,idim)
   ```

   `grades` is a float array containing the grades. `coordinates` is a float array arranged In the 3_D case as $(x_1, y_1, z_1, \ldots, x_{nd}, y_{nd}, z_{nd})$. `nd` is the number of data, and `idim` is the geometric dimension (1, 2 or 3).

2. The program `form_data.f` can do the job when the following restrictions apply:

   - The data at different supports must be stored in separate formated file. This file must contain a data at a line. each line containing the coordinates first and the data after.
   - The coordinates must be coded in the input file as reals.
   - The data are supposed to be numbers. Hence they must be coded at integer format.
   - These numbers are converted to grades before storage. The user must provide the sample volume to allow this conversion.

   See the case study for a sample run of this program. This program is to be called for each data support.

Then the data must be arranged into a block grid. As in the disjunctive kriging programs, the blocks are grouped into rectangular panels. and the kriging neighborhoods are defined in terms of panels. This means that all the blocks within a given panel will have the same kriging neighborhood.

For example in the 2-D case, if the panel consits of $2 \times 4$ blocks. and if the kriging neighborhood is made of $5 \times 3$ panels. the size of the neighborhood in terms of blocks

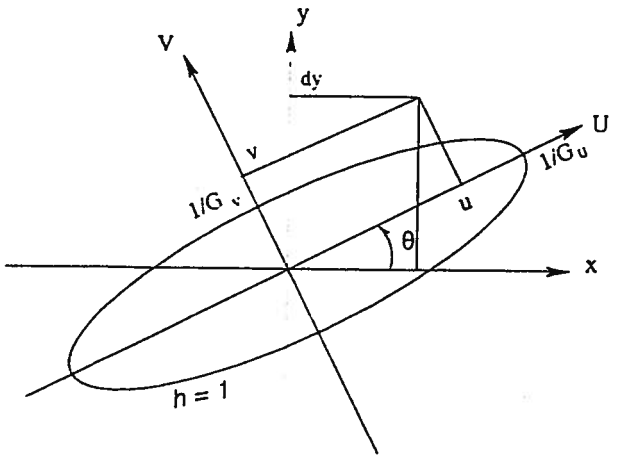



Figure 1: *Convention to define the geometric anisotropies in 2D. The ellipsoid displays the curve $h = 1$ on the Euclidean plane. It intersects the $U$ and $V$ axis at $u = 1/G_u$ and $v = 1/G_v$ respectively.*

is $10 \times 12$. Thus the maximum distance for which the block covariances have to be calculated is $h = (9\,\tau_x, 11\,\tau_y)$ where $\tau_x, \tau_y$ is the block size.

The program `init_f_2D` which is similar to `file_init_2D` for the disjunctive kriging, ensures the setup of the block, of the panel and of the neighborhood structure in the 2-D case. The 3-D case is handled similarly by the program `init_f_3D`. All the sample data at whatever support will be stored in the block file at this stage.

The block grid must also be parallel to the coordinate axis. If a different orientation is needed, it is necessary to change the coordinate system first.

## 2.2 Covariance calculation.

The block covariances at the distances involved in the kriging neighborhood have to be calculated. The variances at all the sample supports present in the data also have to be known and are calculated by the program. All this calculation is done starting from the point grade variogram. The point variogram model is defined interactively by the user. Three questions relative to the definition of the point variogram model deserve comments. These are :

**The nugget effects.** They result from a Dirac component on the point structure. The rule for defining the mass associated with this Dirac component is as follows. If the observed variogram at support $v$ has a nugget effect $\sigma_v^2$, then the Dirac mass at point support is $\sigma_v^2 \times |v|$. Details on this are given in the appendix of this text.

**The anisotropies.** Each basic variogram component is allowed to have its own geometric anisotropy. The anisotropy for a basic structure is defined in 2D by an angle and two coefficients. The angle $\theta$ is the rotation to go from the $ox$ axis to the leading anisotropy direction $U$ (trigonometric convention). This rotation defines a new coordinate system, with axis $U$ and $V$. The two anizotropy coefficients $G_u$ and $G_v$ are range reduction coefficients. If two points are at distance

$(dx, dy)$ on the original coordinate system, the covariance between these two for a structure of covariance $C$ with range $a$ and sill $\sigma^2$ can be obtained by:

- Rotation to get the distance vector in the $U, V$ coordinates :

$$\begin{cases} u &= \cos(\theta)\, dx \;+\; \sin(\theta)\, dy \\ v &= -\sin(\theta)\, dx \;+\; \cos(\theta)\, dy \end{cases}$$

- Calculation of the effective distance :

$$h \;=\; \sqrt{(u\, G_u)^2 + (v\, G_v)^2}$$

- Covariance evaluation :

$$\mathrm{Cov} \;=\; \sigma^2\, C(h/a)$$

The 3-D case is explained in the appendix C.

**The point variogram** should be a grade variogram, even in the case where the data stored in the block file are numbers of particles (number of particles are not meaningful at point support).

The program `covar` does this calculation.

### 2.3 Kriging.

The kriging program is `krb_mxs`. It performs the estimation on each block of the grid (provided that the neighborhood contains at least one data). The output are stored on a file named `krig.res` which contains the three block coordinates, the estimated grade or the estimated number of particles, and the associated estimation variances.

All the calculations are done internally in terms of grades. The conversion back to the number of particles is done in the very end if required. In case of numerical problems during the calculation, an error message is given, and a file is generated which contains the kriging system, and the kriging weights. Its name is `trace`.

## 3 Case study.

The example to illustrate the use of the programs is in 3-D. It assumes that two sample sizes exist, one with a support $0.9 \times 0.9 \times 0.9\ m^3$, the other with a support $3 \times 3 \times 3\ m^3$. The block size will be assumed to be $10 \times 10 \times 10\ m^3$, and the panels $20 \times 20 \times 10\ m^2$. The whole field will be made of $2 \times 2 \times 2$ panels. This geometry and the sample data are shown in figure 2.

### 3.1 Preparing the data structure.

The sample data are in two separate file, `DATA1.ECH` for the support $(0.9)^3\ m^3$ and `DATA2.ECH`, for the support $(3)^3\ m^3$. The program `form_data` ensures the conversion to the appropriate data structure. The sample run shown under is relative to the smaller support.

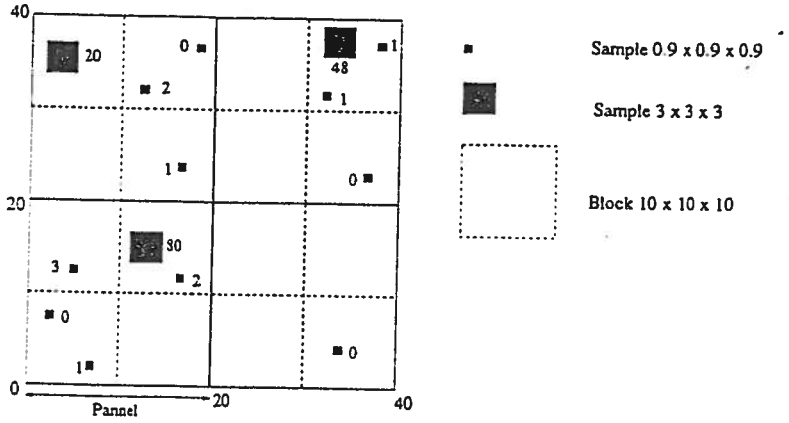



Figure 2: *The data, the block grid, and the panels for the fisrt level:* $0 < z \leq 10\ m.$

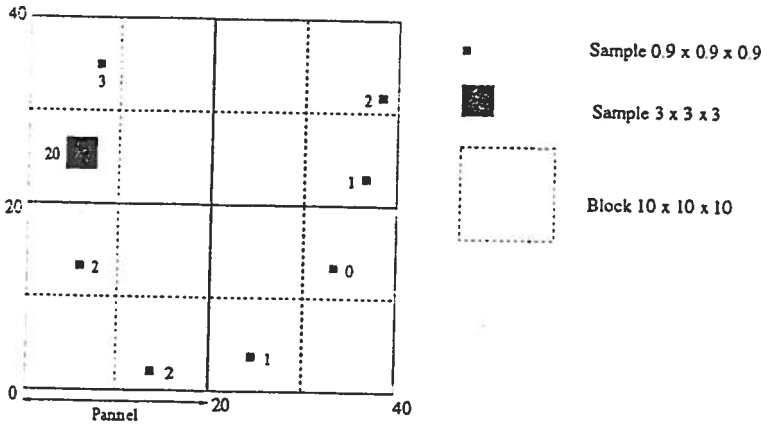
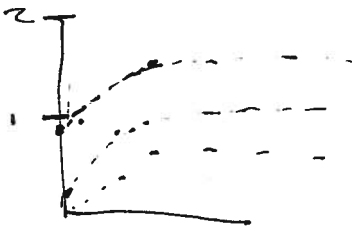



Figure 3: *Second level:* $10 < z \leq 20\ m.$

```
Formated data file ?                                    >DATA1.ECH
Geometric dimension ?                                   >3
Input format (3(Real), Integer) ?
>(f7.2,1x,f7.2,1x,f7.2,i4)
Data support (for conversion to grades) ?               >0.729
Name of the output file ?                               >data1
```

The new data files are now data1 and data2. Then the panel, block, and neighborhood files are prepared by the program init_f_3D, as the following sample run shows :

```
DEFINING THE BLOCK SIZE :
 Along x     ?                                          >10.
 along y     ?                                          >10.
 and along z ?                                          >10.
DEFINING THE FIELD
Origin along x (= x min)?                               >0.
And maximum along x ?                                   >40.
Origin along y ?                                        >0.
And maximum along y ?                                   >40.
Origin along z ?                                        >0.
And maximum along z ?                                   >20.
Number of blocks along x :   4 along y :   4 and along z :   2
Number of blocks in a panel along x ?                   >2
 along y ?                                              >2
 and along z ?                                          >1
Number of panels along x :   2 along y :   2 and along z :   2
ACQUIRING DATA
Number of supports to be considered (max= 3) ?   >2
Reading the sample data for the support  1
Input data file ?                                       >data1
    18 Valid data points read
Reading the sample data for the support  2
Input data file ?                                       >data2
     4 Valid data points read
Number of data for support 1 :   18 Mean =0.1677E+01
Number of data for support 2 :    4 Mean =0.1556E+01
Number of informed blocks =    17
File for the block structure ?                          >block.dat
File to contain the panel structure ?                   >panel.dat
DEFINING THE NEIGHBOURHOOD SIZE (in number of panels)
Number of panels along x ?                              >3
 (Odd integer leq to  7)
 along y ? (max =  7)                                   >3
 and along z ? (max =  7)                               >3
Name of the neighbourhood file ?                        >neigh.dat
```



| Support | $0.9 \times 0.9 \times 0.9$ | $3 \times 3 \times 3$ | $10 \times 10 \times 10$ |
|---|---|---|---|
| Nugget | 0.892 | 0.024 | 0.0006 |
| Exponential | 0.695 | 0.620 | 0.43 |

Table 1: *Nugget effects resulting from a Dirac mass $\sigma^2 = 0.65$ at point support, and variances associated with the point exponential structure.*

The number of data points reported after reading the file is the total number of valid data points read. The second one given together with the mean is the number of those points falling in the block structure. As the data are converted to grades, the means obtained at various supports should be comparable. The neighborhood created in this example consists of $6 \times 6 \times 3$ blocks.

A listing of this data structure can be obtained by runing the program `form_p`.

### 3.2 Covariance calculation.

The point structure is assumed to be the sum of a Dirac component with mass $\sigma^2 = 0.65$, and of the following exponential covariance :

$$C(h) = 0.73 \exp\{-\frac{h}{12}\}$$

The variances associated with these two model components, for the various supports involved, are shown in table 1. The covariances are calculated by the program `covar`, as shown in the following sample run :

```
Geometric dimension (Integer) ?                          >3
DEFINITION OF THE POINT GRADE VARIOGRAM MODEL :
DIRAC component ?
(the value of the Dirac component is the NUGGET
effect for a UNIT SUPPORT)                 (Real)        >.65
Number of structures ? (Integer)                         >1
Definition of the structure number :   1
Type (Int - help=-1) ?                                   >1
Sill (Real) ?                                            >.73
Range (Real) ?                                           >12.
Do you want to define anisotropies
 (yes=1   default=0=no) ?                                >0
Total point variance (excluding the Dirac component) =        0.730000
Block size along the x axis ? (Real)                     >10.
Number of blocks in the neighborhood along
 this direction ?                                        >6
Block size along the y axis ? (Real)                     >10.
Number of blocks in the neighborhood along
 this direction ?                                        >6
Block size along the z axis ? (Real)                     >10.
Number of blocks in the neighborhood along
```

Note: Unit support

```
 this direction ?                                    >3
Number of different data supports to consider
          (maximum =  3) ?                           >2
Defining the sample volume for the support  1
Dimension of the sample along the x direction       >.9
Dimension of the sample along the y direction       >.9
Dimension of the sample along the z direction       >.9
Volume of the sample =     0.7290
Defining the sample volume for the support  2
Dimension of the sample along the x direction       >3.
Dimension of the sample along the y direction       >3.
Dimension of the sample along the z direction       >3.
Volume of the sample =    27.0000
```

Note : The number of blocks in the neighborhood is defined by the parameters of the neighborhood, and not by the domain itself. It is the product of the number of panels in the neighborood $\times$ the number of blocks per panel ($3 \times 2$ in the $x$ direction in this case) even though due to the limited size of the domain it happens that only 4 blocks would be sufficient.

The output of this program, to be used by the kriging program, is on file `cij.dat`. The type 1 above is the exponential, and in this case the parameter range is the scale factor 12. The maximum number of different sample sizes is fixed by a parameter statement on the program (`nsupport_max=3`). Note also that the supports are defined as rectangular (in case of a circular support, the dimension of a rectangular support having the same volume should be given).

It is important that the order in which the different support are given match (they must be given in the same order in `init_f` and in `covar`, thus $0.9 \times 0.9 \times 0.9$ is relative to `data1` and $3 \times 3 \times 3$ to `data2`).

### 3.3 Kriging.

The random kriging program is `krb_mxs`.

```
DEFINING THE GEOMETRY
Panel file ?                                        >pan.dat
Neighbourhood file ?                                >neigh.dat
Block file ?                                        >block.dat
Data = grades (default) or numbers (1) ?            >1
File containing the covariances Cij ?               >cij.dat
Panel number :          1
Panel number :          2
.....
Number of estimated blocks =     32
Conversion of results to numbers of stones ? (yes=1, def=no)
                                                    >
```

The results are stored on the file `krig.res`. If the conversion to number of stones is required, the kriging variances are expressed in squared numbers of particles. The

output format is shown above. It contains the 3 coordinates of the block center (0 for z in the 2D case), the estimated number of particles (or grades), the kriging variances and the slope of the regression $Z/Z^*$:

| X | Y | Z | N* | Var(N-N*) | Slope(N\|N*) |
|---|---|---|---|---|---|
| 5.00 | 5.00 | 5.00 | 1704 | 217853 | 0.8358445 |
| 15.00 | 5.00 | 5.00 | 1964 | 239557 | 0.8406637 |
| 5.00 | 15.00 | 5.00 | 2120 | 186459 | 0.9080035 |
| 15.00 | 15.00 | 5.00 | 2378 | 109118 | 0.9609814 |
| 25.00 | 5.00 | 5.00 | 1573 | 293264 | 0.7757188 |

.....

Because the neighborhood is defined in init_f_3D, it is necessary to run it another time if an increase of the neighborhood is required in order to improve the slope of the regression.

The flow chart of the programs is shown on figure 4.

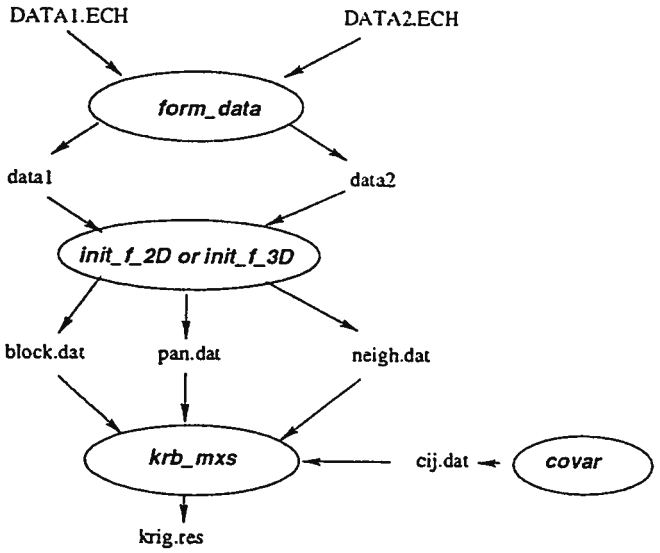



Figure 4: *Flow-chart of the mixed support random kriging programs for the case study. This example contains 2 data support.*

## APPENDICES

# A Conditional bias and the slope of the regression $Z/Z^*$

The conditional bias to be discussed in this section is defined from the bivariate distribution $Z/Z^*$ by the following formula:

$$b(z^*) = E[Z|Z^* = z^*] - z^*$$

It is desirable to keep it as small as possible by an appropriate choice of the estimator $Z^*$. To illustrate the point, if $b(z^*) = 0$ at every $z^*$, then we have:

$$E[Z\, 1_{Z^*>z}] = E[E[Z|Z^*]\, 1_{z^*>z}] = E[Z^*\, 1_{z^*>z}]$$

which means that the actually recovered quantity of metal when the selection is done on the basis of $Z^*$ (the left hand side term), is exactly equals to the production forecast at the mining stage, which is calculated from $Z^*$ alone (the right hand side term).

If the bivariate distribution $Z, Z^*$ were known, it would be possible to correct the conditional bias, or to calculate directly the recovered quantity of metal $E[Z\,1_{Z^*>z}]$. But in general in the context of linear geostatistics this distribution is unknown, and we only know the covariances. Thus we shall limit the discussion to the linear regression, and shall obtain necessary but not sufficient conditions for conditional unbiasedness. This linear regression is defined by:

$$Z^r = m(1-a) + aZ^* + R$$

where $Z^r$ is the linear estimate of $Z$ from $Z^*$, the coefficient $a$ is $\mathrm{Cov}(Z, Z^*)/\mathrm{Var}(Z^*)$, the mean $m = E[Z]$, and $R$ is the residual. If the slope of this regression $a$ is close to one, then the conditional bias will be small, at least as long as the bivariate distribution $Z, Z^*$ is not too far from a normal one.

If the grade estimator is linear, $Z^* = \sum \lambda_i Z_i$ then as long as the kriging weigths are not all zeros [1]:

$$a = \frac{\sum \lambda_i C_{i0}}{\sum \lambda_i \lambda_j C_{ij}}$$

We observe first that if the mean $m$ were known, then the simple kriging could be used, for which $\sum_j \lambda_j C_{ij} = C_{i0}$, and $a = 1$. Thus, the problem comes from the need to estimate the mean. The estimation of the mean by kriging is steadily improving if we consider larger and larger kriging neighborhoods (under mild assumptions $m^* \to m$), so that we can expect to improve the $a$ parameter of the grade ordinary kriging by increasing the size of the neighborhood. This slope can be calculated from the formula [2]:

$$a = \frac{1}{1 - \mu/(\sum C_{i0}\lambda_i)}$$

If the value obtained is too small, $a$ can be made closer to one by increasing the kriging neighborhood. This is shown for the case study presented in this document on Figure 5, where the slopes obtained for each block under a $1 \times 1$ panel neighborhood and under a $3 \times 3$ one.

# B The nugget effect.

It was assumed in this document that a point variogram model is known, from the analysis of the sample variograms at different supports. The question of how to define the nugget effect for the point support arises. We start by recalling a general variance formula. Let $\phi$ and $\psi$ be two functions[3], and let $Z(\phi)$ and $Z(\psi)$ be the following stochastic integrals:

$$Z(\phi) = \int \phi(x)\, Z(x)\, dx$$
$$Z(\psi) = \int \psi(x)\, Z(x)\, dx$$

[1] The simple kriging under a pure nugget model is excluded from this discussion

[2] The sign of the lagrange multiplier $\mu$ is conventional. The formula is consistent with the kriging system $\sum_j C_{ij}\lambda_j + \mu = C_{i0}$.

[3] The validity of the formula holds under some restrictions on $\phi$ and $\psi$. We shall admit that it holds for block indicators.

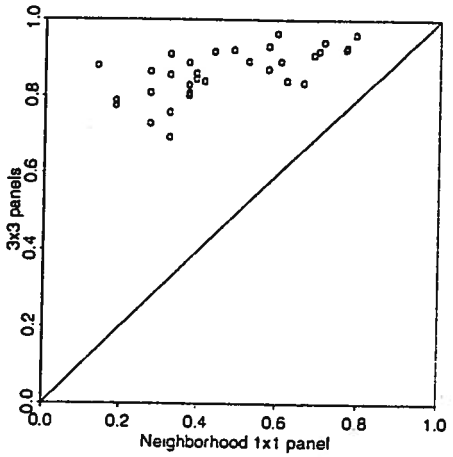



Figure 5: *Slopes of linear regressions for 2 neighborhood sizes.*

If $C$ is the (point stationary) covariance of $Z$, then the covariance of $Z(o)$ and $Z(v)$ can be obtained :

$$\mathrm{Cov}\{Z(o), Z(v)\} = < C, o * \check{v} >$$

In the above formula, "$*$" is the convolution product, defined by :

$$(f * g)(h) = \int f(x)\, g(h - x)\, dx$$

$\check{\psi}$ is defined by $\check{\psi}(x) = \psi(-x)$, and :

$$(\phi * \check{\psi})(h) = \int \phi(x)\, \psi(x - h)\, dx.$$

On the other hand,

$$< C, f > = \int C(h)\, f(h)\, dh$$

Combining these two, and changing the integration variables, we get the following explicit expression :

$$< C, \phi * \check{\psi} > = \int\int C(h)\, \phi(x)\, \psi(x - h)\, dx\, dh = \int\int C(x - y)\, \phi(x)\, \psi(y)\, dx dy$$

which is the well known formula.

The advantage of the previous formula is that it remains valid when the structure of $Z$ is defined by a Dirac measure of mass $s^2$, instead of the continuous covariance $C$

$$< s^2\, \delta, \phi * \check{\psi} > = s^2 \int\int \phi(x) \psi(x - h)\, dx\, \delta(dh) = s^2 \int \phi(x)\, \psi(x)\, dx$$

For the indicator functions of two compact sets $A$ and $B$ :

$$\phi(x) = \frac{1}{|A|}\, 1_A(x) \qquad \text{and} \qquad \psi(x) = \frac{1}{|B|}\, 1_B(x)$$

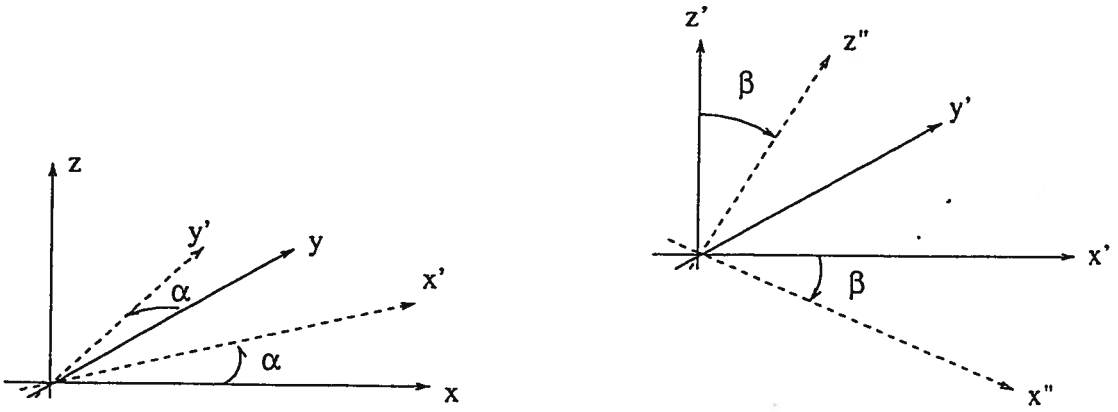



Figure 6: *First and second rotations.*

we have:

$$\mathrm{Cov}\{Z(A), Z(B)\} = s^2 \frac{1}{|A||B|} \int 1_A(x)\, 1_B(x)\, dx = s^2 \frac{|A \cap B|}{|A||B|}$$

To obtain the block variance $\mathrm{Var}(Z(V))$, we consider the case $A = B = V$:

$$\sigma_V^2 = \frac{s^2}{|V|} \qquad \text{and} \qquad \overline{C}(V, V') = 0 \ \ (\text{if } V \cap V' = \emptyset)$$

This gives us the following practical rule that we are looking for [4]:

> The Dirac point covariance with mass $s^2$ gives at the support $V$ a nugget effect with variance
>
> $$\frac{s^2}{|V|}$$

Conversely, if the covariance at support $V$ has a nugget component of variance $\sigma_V^2$, the Dirac mass of the covariance model at point support is $s^2 = \sigma_V^2 \times |V|$.

## C Anizotropies in 3-D

The anizotropies can be defined by specifying three rotations, and three coefficients. The rotations are the following:

- Rotation $\mathcal{R}_1$, with center $o$, axis $oz$, and angle $\alpha$ with trigonometric convention. This changes the axis $ox, oy, oz$ to $ox', oy', oz'$ (Figure 6)
- Rotation $\mathcal{R}_2$ with axis $oy'$, and angle $\beta$ (positive downwards). This changes $ox', oy', oz'$ to $ox'', oy'', oz''$.
- Rotation $\mathcal{R}_3$ with axis $ox''$, and angle $\gamma$ with trigonometric convention. From this $ox'', oy'', oz''$ become $ox''', oy''', oz'''$ (Figure 7)

[4] This statement obviously ignores the behaviour at small distances $h$ such that $V \cap V_h \neq \emptyset$

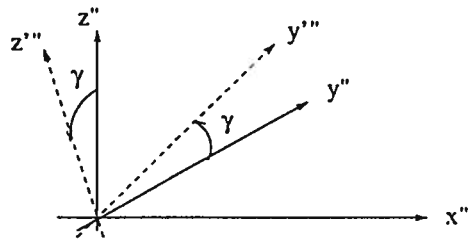



Figure 7: *Thrid rotation.*

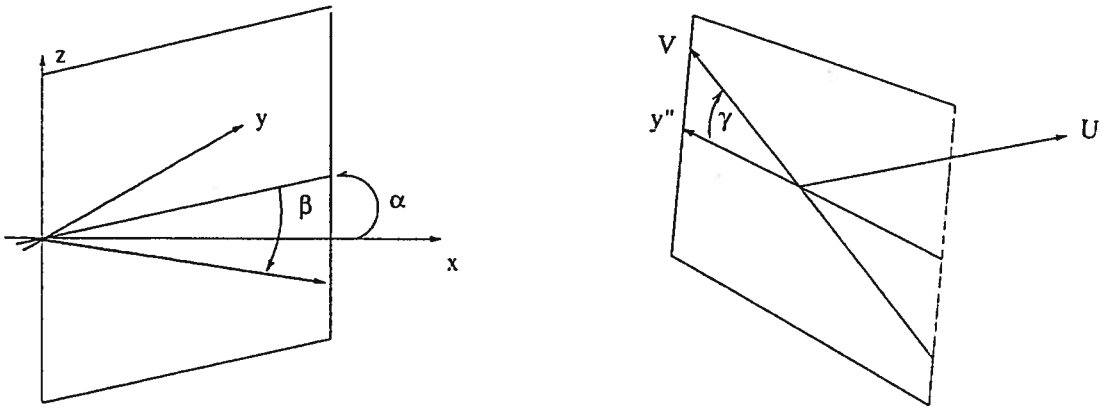



Figure 8: *Definition of the leading and second anisotropy directions.*

The composition of these rotations change the coordinates $(dx, dy, dz) - (u, v, w)$, and the distance to be used in the variogram is:

$$h = \sqrt{(uG_u)^2 + (vG_v)^2 + (wG_w)^2}$$

## C.1 definition in terms of anisotropy ellipsoid

It is intuitively easier to think in terms of leading anisotropy directions. Each of these can be defined either in terms of a unit vector, or in terms of angles. The definition in terms of angles is as follows:

**First axis.** This direction is defined first by the angle between its projection on the horizontal plane, and the $ox$ axis (trigonmetric convention). This is the angle $\alpha$. Then, the second angle $\beta$ is between the axis and the horizontal plane, with positive convention downwards (figure 8).

**Second axis** It is in the plane orthogonal to the first one, and is defined by its angle in that plane with the horizontal line ($\gamma$ is this angle, counted positively upwards).

The third axis is orthogonal to the first two and need not be specified. The tree angles mentioned here are the same as the rotation angles of the previous paragraph.

As an example, suppose that a spherical structure had been identified, with:

- Range 20 in the direction East-North-East with 10 deg plunge.

- Range 10 in the orthognal direction with a rotation +5 deg (approximately orientation North-West-North, 5 deg upwards.)
- Range 5 in the third direction.

Then we can specify a spherical structure with range 20. and anizotropies given by $\alpha = 30, \beta = 10, \gamma = 5$ and $(G_u, G_v, G_w) = (1, 2, 4)$.

## C.2 Calculation of change of coordinates.

This section is intended to help in the programming of anizotropies, but is not necessary to use the programs. We begin with a few linear algebra.

Let $T$ be a linear mapping in $\mathbb{R}^n$. If an orthonormal basis $\epsilon_1, \ldots \epsilon_n$ is specified, a matrix representation of $T$ is defined by assigning to each column the coefficients of the transform of the basis:

$$A = (T(e_1), T(e_2), \ldots, T(\epsilon_n))$$

or in other words, $T(\epsilon_i) = \sum_j a_{ji} e_j$. Now if $M$ is a point and $OM$ the vector with origin $O$ and extreminy $M$. Let the coefficients of $OM$ in $\epsilon_1, \ldots \epsilon_n$ be $x_1, \ldots x_n$, so that

$$OM = \sum_i x_i \, e_i$$

and $(y_1, \ldots y_n)$ be the components of $T(OM)$:

$$T(OM) = \sum_i y_i \, \epsilon_i$$

then

$$T(OM) = \sum_i x_i \, T(\epsilon_i) = \sum_{ij} x_i \, a_{ji} \, \epsilon_j$$

and we identify (the decomposition being unique) $y_j = a_{ji} x_i$. In matrix form:

$$Y = A \, X$$

In other words, the previously defined matrix of $T$ account of the coordinates of the transform of any vector. From this, it is obvious that if $T_1$ and $T_2$ are two linear mappings of $\mathbb{R}^n$ with matrices $A_1$ and $A_2$ (expressed in the same basis) the mapping $T_2 \circ T_1$ has a matrix given by the product $A_2 A_1$.

Now, if $T$ is such that the scalar product is conserved:

$$\forall (OM, OM'), \qquad < T(OM), T(OM') > = < OM, OM' >$$

or in matrix form:

$$\forall (X, X'), \qquad X^t A^t A \, X' = X^t X'$$

If we apply this to the elements of the basis $X = \epsilon_i, X' = \epsilon_j$ we see that:

$$A^t A = I$$

so that clearly $A^{-1} = A^t$. This is the case if $A$ represents a rotation. The image of the orthonormal basis $e_i; i = 1, \ldots n$ by $T$ is another orthonormal basis of $\mathbb{R}^n$. Let $(f_1, .., f_n)$ be this new basis:

$$(f_1, f_2, \ldots f_n) = (T(\epsilon_1), \ldots T(\epsilon_n))$$

Let $OM = \sum_i x_i \epsilon_i$. It is usefull to have a formula to calculate the coefficients of $OM$ in the new basis.

$$OM = \sum_i x'_i f_i$$

The calculation is straighforward:

$$\sum_i x'_i f_i = \sum_i x'_i T(\epsilon_i) = \sum_{ij} x'_i a_{ji} \epsilon_j$$

And we identify:

$$A X' = X \qquad \Longrightarrow \qquad X' = A^{-1} X$$

Thus the coordinates of a given point in the new system are obtained by applying the inverse of the transformation matrix. In case of a rotation, $X' = A^t X$.

Now let $R$ be the matrix of a rotation which change of coordinate system $(\epsilon_1, \ldots, \epsilon_n) \rightarrow (f_1, \ldots f_n))$ and let $T$ be linear mapping with matrix $A'$ in the system $\{f_i\}$. We want to calculate the matrix of $T$ in the system $\epsilon_i$. For any vector $OM$ we have:

$$OM = \sum u_i f_i = \sum x_i \epsilon_i \qquad T(OM) = \sum v_i f_i = \sum y_i \epsilon_i$$

We know that $V = A'U$. But $U = R^t X$ and $V = R^t Y$ and:

$$Y = R A' R^t X$$

hence the matrix of $T$ in the system $\{\epsilon_i\}$ is $A = RA'R^t$.

## C.3 Application to anizotropies

We compose three elementary rotations:

**First rotation:** Rotation in the plane $xoy$, which move the axis to $x'y'z'$. The matrix of this transform is:

$$R_1 = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

**Second rotation:** Rotation in the plane $x'oz'$. which move the axis to $x''y''z''$. The matrix of this transform in the coordinate system of $x'y'z'$ is:

$$R_2 = \begin{pmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{pmatrix}$$

Hence its matrix in the original coordinate system is $R_1 R_2 R_1^t$. and the composition of the first two rotations is represented in this system by the matrix:

$$(R_1 R_2 R_1^t) R_1 = R_1 R_2.$$

**Third rotation:** Rotation in the plane $y''oz''$, which move the axis to $x'''y'''z'''$. The matrix of this transform (in the system of $x''y''z''$) is:

$$R_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\gamma) & -\sin(\gamma) \\ 0 & \sin(\gamma) & \cos(\gamma) \end{pmatrix}$$

In the original system the associate matrix is:

$$(R_1\, R_2)\, R_3\, (R_1\, R_2)^t$$

The matrix of the composition of the three rotations $R_1\, R_2\, R_3$ is after calculation:

$$\begin{pmatrix} \cos(\alpha)\cos(\beta) & -\sin(\alpha)\cos(\gamma) + \cos(\alpha)\sin(\beta)\sin(\gamma) & \sin(\alpha)\sin(\gamma) + \cos(\alpha)\sin(\beta)\cos(\gamma) \\ \sin(\alpha)\cos(\beta) & \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\beta)\sin(\gamma) & -\cos(\alpha)\sin(\gamma) + \sin(\alpha)\sin(\beta)\cos(\gamma) \\ -\sin(\alpha) & \cos(\beta)\sin(\gamma) & \cos(\beta)\cos(\gamma) \end{pmatrix}$$

and if a vector has coordinates in the first system $h$, its coordinates in the system given by the anizotropies is $h' = R^t\, h$.

## References

[1] Georges Matheron. *Les Variables Régionalisées et leur estimation.* Masson, 1965.

# Mixed Support Kriging

What follows is an explanation of Lajaunie [1, §1.1]. As in that article, we assume that the block to be kriged is denoted by $V$ and the blocks within the kriging neighbourhood are $V_k, k = 1, \ldots, N$. Samples within $V_k$ of the same support $v_i$ will be written as $Z(v_{k,i}^{\alpha}), \alpha = 1, \ldots, n_k^i$.

In order to determine the mixed support random kriging equations, we first need to determine the variance, $\sigma_\epsilon^2$, of the error of estimation

$$\epsilon = Z_V^* - Z_V,$$

where

$$Z_V^* = \sum_k \sum_i \lambda_k^i \sum_{\alpha=1}^{n_k^i} Z(v_{k,i}^{\alpha}). \tag{1}$$

In the kriging estimator of (1), samples of the same support within a given block are assigned the same weight. In addition, the kriging weights, $\{\lambda_k^i\}$, are taken to satisfy the universality condition,

$$\sum_k \sum_i n_k^i \lambda_k^i = 1. \tag{2}$$

Now

$$\begin{aligned} \sigma_\epsilon^2 &= \mathbb{V}(Z_V^* - Z_V) \\ &= \mathbb{V}\left[\sum_k \sum_i \lambda_k^i \sum_{\alpha} Z(v_{k,i}^{\alpha}) - \sum_k \sum_i n_k^i \lambda_k^i Z_V\right] \end{aligned} \tag{3}$$

where (3) is obtained using (1) and (2).

Next, from (3), we have

$$\begin{aligned} \sigma_\epsilon^2 &= \mathbb{V}\left[\sum_k \sum_i \lambda_k^i \sum_{\alpha} \left(Z(v_{k,i}^{\alpha}) - Z_V\right)\right] \\ &= \sum_k \sum_i \sum_{k'} \sum_{i'} \lambda_k^i \lambda_{k'}^{i'} \sum_{\alpha=1}^{n_k^i} \sum_{\alpha'=1}^{n_{k'}^{i'}} \operatorname{cov}\left[Z(v_{k,i}^{\alpha}) - Z_V, Z(v_{k',i'}^{\alpha'}) - Z_V\right] \end{aligned}$$

Note that

$$\begin{aligned} &\operatorname{cov}\left[Z(v_{k,i}^{\alpha}) - Z_V, Z(v_{k',i'}^{\alpha'}) - Z_V\right] \\ = \; &\mathbb{V}(Z_V) + \operatorname{cov}[Z(v_{k,i}^{\alpha}), Z(v_{k',i'}^{\alpha'})] - \operatorname{cov}[Z(v_{k,i}^{\alpha}), Z_V] - \operatorname{cov}[Z(v_{k',i'}^{\alpha'}), Z_V]. \end{aligned} \tag{4}$$

Hence we can write

$$\sigma_\epsilon^2 = S_1 + S_2 - S_3 - S_4, \tag{5}$$

where the $\{S_j\}$ are defined below.

$$\begin{aligned} S_1 &\triangleq \sum_k \sum_i \sum_{k'} \sum_{i'} \lambda_k^i \lambda_{k'}^{i'} \sum_{\alpha=1}^{n_k^i} \sum_{\alpha'=1}^{n_{k'}^{i'}} \mathbb{V}(Z_V) \\ &= \left( \sum_k \sum_i n_k^i \lambda_k^i \right) \left( \sum_{k'} \sum_{i'} n_{k'}^{i'} \lambda_{k'}^{i'} \right) \sigma_V^2 \\ S_1 &= \sigma_V^2, \end{aligned} \tag{6}$$

where the last line is a result of (2).

The second component of $\sigma_\epsilon^2$ is

$$\begin{aligned} S_2 &\triangleq \sum_k \sum_i \sum_{k'} \sum_{i'} \lambda_k^i \lambda_{k'}^{i'} \sum_{\alpha=1}^{n_k^i} \sum_{\alpha'=1}^{n_{k'}^{i'}} \operatorname{cov}[Z(v_{k,i}^\alpha), Z(v_{k',i'}^{\alpha'})] \\ &= \sum_{k=k'} \sum_{i=i'} \sum_{\alpha=\alpha'} \lambda_k^i \lambda_{k'}^{i'} \operatorname{cov}[Z(v_{k,i}^\alpha), Z(v_{k',i'}^{\alpha'})] \\ &\quad + \sum_{k=k'} \sum_{i=i'} \sum_{\alpha' \neq \alpha} \lambda_k^i \lambda_{k'}^{i'} \operatorname{cov}[Z(v_{k,i}^\alpha), Z(v_{k',i'}^{\alpha'})] \\ &\quad + \sum_{k=k'} \sum_{i' \neq i} \sum_{\alpha,\alpha'} \lambda_k^i \lambda_{k'}^{i'} \operatorname{cov}[Z(v_{k,i}^\alpha), Z(v_{k',i'}^{\alpha'})] \\ &\quad + \sum_{k' \neq k} \sum_{i,i'} \sum_{\alpha,\alpha'} \lambda_k^i \lambda_{k'}^{i'} \operatorname{cov}[Z(v_{k,i}^\alpha), Z(v_{k',i'}^{\alpha'})] \\ S_2 &= \sum_k \sum_i n_k^i \left(\lambda_k^i\right)^2 \sigma_i^2 \\ &\quad + \sum_k \sum_i n_k^i (n_k^i - 1) \left(\lambda_k^i\right)^2 \sigma_V^2 \\ &\quad + \sum_k \sum_{i' \neq i} n_k^i n_k^{i'} \lambda_k^i \lambda_k^{i'} \sigma_V^2 \\ &\quad + \sum_{k' \neq k} \sum_{i,i'} n_k^i n_{k'}^{i'} \lambda_k^i \lambda_{k'}^{i'} \overline{C}(V_k, V_{k'}). \end{aligned} \tag{7}$$

The third component from (5) is

$$\begin{aligned} S_3 &\triangleq \sum_k \sum_i \sum_{k'} \sum_{i'} \lambda_k^i \lambda_{k'}^{i'} \sum_{\alpha=1}^{n_k^i} \sum_{\alpha'=1}^{n_{k'}^{i'}} \mathrm{cov}[Z(v_{k,i}^{\alpha}), Z_V] \\ &= \sum_k \sum_i n_k^i \lambda_k^i \left( \sum_{k'} \sum_{i'} n_{k'}^{i'} \lambda_{k'}^{i'} \right) \overline{C}(V_k, V) \\ S_3 &= \sum_k \sum_i n_k^i \lambda_k^i \overline{C}(V_k, V), \end{aligned} \tag{8}$$

where (2) is again used in obtaining the last line.

Finally, we have

$$\begin{aligned} S_4 &\triangleq \sum_k \sum_i \sum_{k'} \sum_{i'} \lambda_k^i \lambda_{k'}^{i'} \sum_{\alpha=1}^{n_k^i} \sum_{\alpha'=1}^{n_{k'}^{i'}} \mathrm{cov}[Z(v_{k',i'}^{\alpha'}), Z_V] \\ S_4 &= S_3. \end{aligned} \tag{9}$$

A combination of (4), (5) and (6)–(9) yields

$$\begin{aligned} \sigma_\epsilon^2 = \; & \sigma_V^2 + \sum_k \sum_i [n_k^i \sigma_i^2 + n_k^i (n_k^i - 1) \sigma_V^2] \left(\lambda_k^i\right)^2 \\ & + \sum_k \sum_{i' \neq i} n_k^i n_k^{i'} \sigma_V^2 \lambda_k^i \lambda_k^{i'} + \sum_{k' \neq k} \sum_{i,i'} n_k^i n_{k'}^{i'} \overline{C}(V_k, V_{k'}) \lambda_k^i \lambda_{k'}^{i'} \\ & - 2 \sum_k \sum_i n_k^i \overline{C}(V_k, V) \lambda_k^i \end{aligned} \tag{10}$$

as the variance of the estimation error.

The kriging equations are derived by minimising $f(\boldsymbol{\lambda}, \mu) = \sigma_\epsilon^2 - 2\mu \left(1 - \sum_{k,i} n_k^i \lambda_k^i\right)$ w.r.t. the $\{\lambda_k^i\}$ and the Lagrangian multiplier, $\mu$. Since

$$\begin{aligned} \frac{\partial f(\boldsymbol{\lambda}, \mu)}{\partial \lambda_r^j} = \; & 2[n_r^j \sigma_j^2 + n_r^j (n_r^j - 1) \sigma_V^2] \lambda_r^j + 2 \sum_{j' \neq j} n_r^j n_r^{j'} \sigma_V^2 \lambda_r^{j'} \\ & + 2 \sum_{r' \neq r} \sum_{j'} n_r^j n_{r'}^{j'} \overline{C}(V_r, V_{r'}) \lambda_{r'}^{j'} - 2 n_r^j \overline{C}(V_r, V) + 2\mu n_r^j \lambda_r^j, \end{aligned}$$

the mixed support random kriging equations are

$$\begin{aligned}[n_k^i\sigma_i^2 + n_k^i(n_k^i-1)\sigma_V^2]\lambda_k^i + \sum_{i'\neq i} n_k^i n_k^{i'}\sigma_V^2\lambda_k^{i'} \\ + \sum_{k'\neq k}\sum_{i'} n_k^i n_{k'}^{i'}\overline{C}(V_k,V_{k'})\lambda_{k'}^{i'} + n_k^i\mu &= n_k^i\overline{C}(V_k,V), \quad \forall\, k,i \qquad (11)\\ \sum_k\sum_i n_k^i\lambda_k^i &= 1.\end{aligned}$$

If we multiply (11) through by $\lambda_k^i$ and then sum the resulting equations over $k$ and $i$, it emerges from (10) that the kriging variance, $\sigma^2_{\text{MSK}}$, is

$$\begin{aligned}\sigma^2_{\text{MSK}} &= \sigma_V^2 + \left[\sum_{k,i} n_k^i\overline{C}(V_k,V)\lambda_k^i - \mu\sum_{k,i} n_k^i\lambda_k^i\right] - 2\sum_k\sum_i n_k^i\overline{C}(V_k,V)\lambda_k^i \\ &= \sigma_V^2 - \sum_{k,i}[\overline{C}(V_k,V)+\mu]n_k^i\lambda_k^i.\end{aligned}$$

# References

[1] C. Lajaunie. Documentation of the mixed support kriging programs, 1996.