

A Global Kriging Program for Artifact-Free Maps

Chad Neufeld and Brandon Wilde

Centre for Computational Geostatistics
Department of Civil & Environmental Engineering
University of Alberta

Artifacts are common in almost all kriged maps when a limited search is used. A solution is to use all of the samples for kriging all of the locations; however, this can be very slow as the system of kriging equations is solved for each location. A global kriging program was written to use all of the sample data for the estimation to prevent these artifacts from occurring. This allows all the samples to be used for generating the kriged maps. In addition, the CPU time tends to decrease dramatically.

Introduction

Geostatistics is concerned with modeling the spatial distribution of natural attributes such as mineral grades, petrophysical properties, or contaminant concentrations. The most common method used for modeling these attributes is kriging. Kriging is commonly called the best linear unbiased estimator. It is the only estimation method that can simultaneously account for the spatial correlation between the data and the point being estimated and the redundancy between data values.

Kriging is accomplished by first going to a location where the attribute is unknown, searching to identify the nearest data, solving the system of equations to get weights for each of the sample data, calculating the estimate using the weights and the samples. The user must specify a search radius and a maximum number of data to be used for the estimate. These subjective parameters create artifacts in the kriged estimates. These artifacts have no basis geologically, but are completely due to the parameters used. Ordinary kriging exaggerates these artifacts compared to simple kriging. To avoid these artifacts, a simple solution is to use an unlimited number of data and an exhaustive search. However, the typically kriging programs will solve this large system of equations for each location being estimated, causing the CPU time to increase substantially. If all the samples are used for estimating all of the locations, a different kriging approach can be used that substantially reduces the amount of CPU time. This is referred to as global kriging.

Global kriging does not produce any artifacts of the search, because no search is used. The program reads in all of the data and uses all of the data values for estimating each location. An additional benefit of global kriging is reduced CPU time for calculating all of the estimates. The methodology used for the global kriging approach is explained in the following section, and a short example completes the paper.

Methodology

Kriging is a linear estimator of an attribute at a given location. The weights assigned to the samples are calculated based on the spatial relationship between the data values and the point being estimated and the data values and themselves. Kriging is done as follows:

1. Go to a location where the value of the attribute is to be estimated,

2. Perform a search to identify the nearby data that will be considered. We search until we find the maximum number of data or until we reach the maximum search radius.
3. Calculate the covariance between the data points that are being used as well as the covariance between the data and the location being estimated.
4. Solve the kriging equations to get the weights.
5. Use the weights to calculate the estimate.

This process is repeated for all locations being estimated; often for hundreds of thousands or millions of locations. Creating all of these estimates is very time consuming, especially as more data points are considered. Using a limited search creates artifacts in the resulting estimates. The only way to avoid these artifacts is to do away with the limited search and its parameters. But without them there is no way to limit the number of data used. With unlimited data, the estimation can take a very long time. The key to the global kriging program was the recognition that the large left-hand matrix will be the same for all locations being estimated if all of the data are considered for every location. We only need to invert that matrix once.

A kriged estimate is a linear combination of the surrounding data values:

$$Z^*(\mathbf{u}) = \sum_{i=1}^n \lambda_i \cdot Z(\mathbf{u}_i) \quad (1)$$

where \mathbf{u} refers to a location, $Z^*(\mathbf{u})$ is the estimate at location \mathbf{u} , there are n data values, and λ_i is the weight assigned to the i^{th} data value. These weights can be computed by solving the kriging equations. In this case, the ordinary kriging equations:

$$\begin{bmatrix} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & 1 \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{bmatrix} \quad (2)$$

where C_{ij} is the covariance between data points i and j , C_{i0} is the covariance between the i^{th} data point and location being estimated, n is the number of data points being used, and μ is the lagrange parameter need for ordinary kriging. As can be seen from the in Equation (2), the size of the matrices and vectors is the same as the number of data plus one. The huge system of equations would have to be re-solved for every estimated location, which would be extremely time consuming.

Let

$$[K] = \begin{bmatrix} C_{11} & \cdots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & 1 \\ C_{n1} & \cdots & C_{nn} & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}, [k] = \begin{bmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{bmatrix}, \text{ and } [\lambda] = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \\ \mu \end{bmatrix} \quad (3)$$

We can express Equation (1) as:

$$Z^*(\mathbf{u}) = [\lambda]^T \cdot [Z_\alpha] \quad (4)$$

where

$$[\lambda] = [\lambda_1, \dots, \lambda_n, \mu] \text{ and } [z_\alpha] = \begin{bmatrix} z_1 \\ \vdots \\ z_n \\ 0 \end{bmatrix}$$

The zero is needed so that the array of weights and data values are the same length, and to exclude the lagrange parameter μ from the estimate $Z^*(\mathbf{u})$. We can re-write Equation (2) in matrix terms:

$$[K][\lambda] = [k] \quad (5)$$

The weights can be solved by rearranging the equation to get:

$$[\lambda] = [K]^{-1} [k] \quad (6)$$

Then substitute the weights from Equation (6) back into Equation (4):

$$\begin{aligned} Z^*(\mathbf{u}) &= [\lambda]^T [Z_\alpha] \\ &= \left([K]^{-1} [k] \right)^T [Z_\alpha] \\ &= [k]^T [K]^{-1} [Z_\alpha] \end{aligned} \quad (7)$$

It is interesting to note that the large covariance matrix K does not change from one location to the next if all of the data are used for every location. The traditional kriging algorithms would solve the system of equations for each node being estimated. The global kriging algorithm solves for the inverted matrix K^{-1} and retains it for subsequent estimates. This modification allows all of the samples to be used for creating artifact free maps without a significant increase in CPU time.

This is especially useful with respect to ordinary kriging. Ordinary kriging tends to create artifacts when data are included or excluded from the search neighborhood. This is because ordinary kriging uses a locally varying mean calculated from the sample data used for estimating the location. When data are dropped or included, the mean value changes from one block to the next. Artifacts are also a function of the number of data used to estimate each location. Using more data creates a smoother map. This means that global kriging allows us to create a better map in less time.

Program

The global kriging program (kt3d_g1) follows standard GSLIB conventions. The parameters for the program are shown:

```

Line  START OF PARAMETERS:
  1      cluster.dat           -file with data
  2      0 1 2 0 3 0          -columns for DH,X,Y,Z,var,sec var
  3      -1.0e21  1.0e21      -trimming limits
  4      0                    -debugging level: 0,1,2,3

```

```

5      kt3d_gl.dbg          -file for debugging output
6      kt3d_gl.out         -file for kriged output
7      70 .85 .8           -nx,xmn,xsiz
8      70 .85 .8           -ny,ymn,ysiz
9      1  0.5  1.0         -nz,zmn,zsiz
10     1  1  1             -x,y and z block discretization
11     1  2.302            -0=SK,1=OK,2=LVM[resid],3=LVM,5=exdrift
12     extdrift.dat        -gridded file with drift/mean
13     4                   -column number in gridded file
14     1 0.2               -nst, nugget effect
15     1 0.8 0.0  0.0  0.0 -it,cc,ang1,ang2,ang3
16           10.0 10.0 10.0 -a_hmax, a_hmin, a_vert

```

The input file is specified in **Line 1**. **Line 2** specifies the drillhole, its location, and the variable being considered. The debugging level specified in **Line 4** controls how much information is written to the debug file specified in **Line 5**. **Line 6** specifies the file where the kriging output will be. **Lines 7, 8, and 9** specify the grid in the x, y, and z directions respectively that is to be estimated. The first column indicates the number of grid cells, the second is the centre point of the first block, and the third is the size of the grid cells. The block discretization is specified in **Line 10**. **Line 11** is where the type of kriging is indicated. If there is a file containing the grid, it is specified in **Line 12** with the column given in **Line 13**. **Line 14** details the number of nested structures in the variogram along with the nugget effect. **Lines 15 and 16** specify the axes and ranges for the variogram.

The program output file has two columns. The first is the estimate for every grid cell. The second is the estimation variance for each cell.

Examples

Testing of the global kriging program was done using the cluster.dat data file. There are 140 data in a 50m x 50m area. Figure 1 shows the location of the data from the file cluster.dat. The scale represents the values of the primary variable. Figures 2-4 are pixel plots of kriged estimates using the 140 sample data. Figure 2 shows a kriged model that is 500 cells by 500 cells (each cell is 0.1m square). For the first plot, upper left corner, the kriging was performed using the conventional kriging program limiting the amount of data used to 8. The artifacts caused by this parameter can be plainly seen, especially the large conglomeration of mid-range values along the top edge. The second plot, upper right corner, representing the kriging with 36 data points also shows artifacts, especially noticeable in the upper left corner. The kriging results where all of the samples were used show no artifacts. The bottom left figure was kt3d using all of the sample data and the lower right was the global kriging program.

Figure 3 is a graph of the CPU time required for the estimation versus the number of cells estimated. The different lines represent the different maximum number of samples used. It is easily seen that the more data that is used to calculate an estimate, the more time it takes. It is interesting to note that the global kriging program takes less time to calculate estimates than the conventional kriging program does using even smaller amounts of data.

There is a cross-over point where the global program becomes faster than the conventional program for small amounts of data. Since matrix inversion is numerically challenging, the global kriging programs pays a penalty upfront for inverting the matrix K , but makes up for that penalty as the number of locations estimated increases.

Figures 4-6 show some kriged maps using the `red.dat` data file included with the GSLIB book. Three kriginings were performed, all with the same variogram and data set: (1) an ordinary kriging with a maximum of 8 sample data, Figure 4, (2) an ordinary kriging with a maximum of 8 sample data an octant search with a maximum of 3 samples per octant, Figure 5, and (3) a global kriging where all of the samples were used, Figure 6. The artifacts that are apparent in Figures 4 and 5 are a result of the limited search neighborhood used. The global kriging is much smoother, but there are no artifacts present.

Conclusions

Almost all kriged maps show some artifacts. These artifacts are often due to the limited search neighborhood typically used during kriging. A limited search is required when there are more than 50-100 samples because of CPU time. Global kriging is an effective method for creating artifact-free kriged maps without a severe increase in CPU time. By using all of the sample data, smooth artifact free maps can be generated.

The first limitation of this approach is greater emphasis on the assumption of stationarity. This will have to be evaluated on a case-by-case basis; however, visual inspection of the maps created to date to not show unwarranted smoothing. A second limitation is CPU time with very large sets of data. It would not be advisable to consider more than a few thousand data. Our experience has mostly been with 2-D datasets.

References

- [1] Deutsch, C.V. and Journel, A.G., 1998. *GSLIB: Geostatistical Software Library: and User's Guide*. Oxford University Press, New York, 2nd Ed.

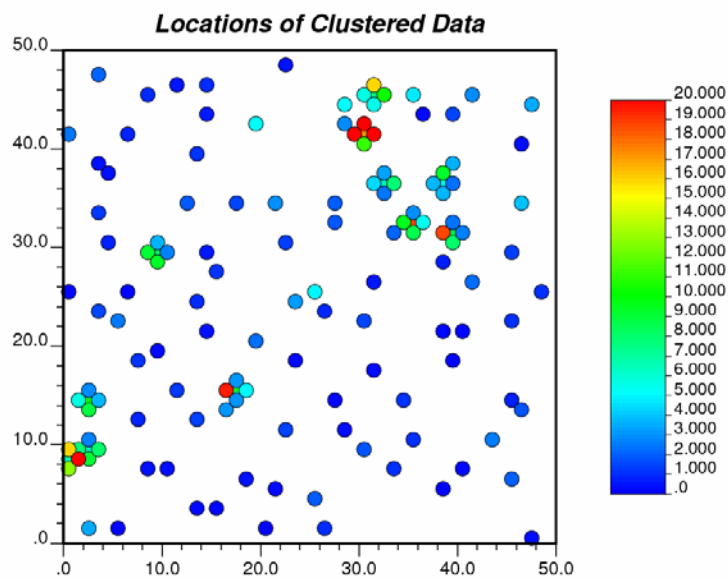


Figure 1: Location of data coded by primary variable values.

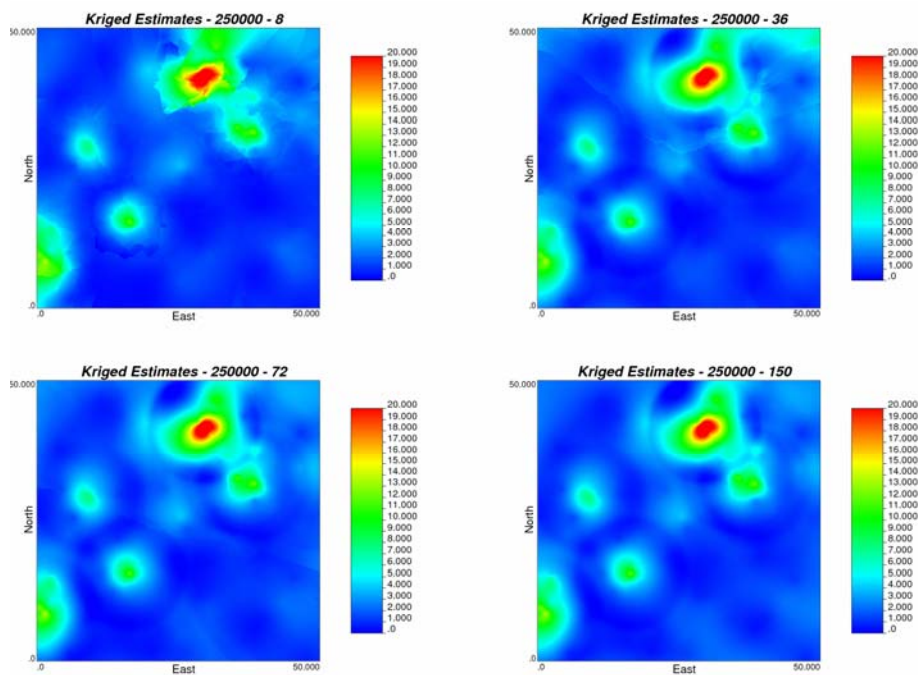


Figure 2: Kriged Estimates of a 500x500 grid. The first 3 are plots of the estimates using 8, 36 and 62 data respectively. The last plot uses the global kriging program.

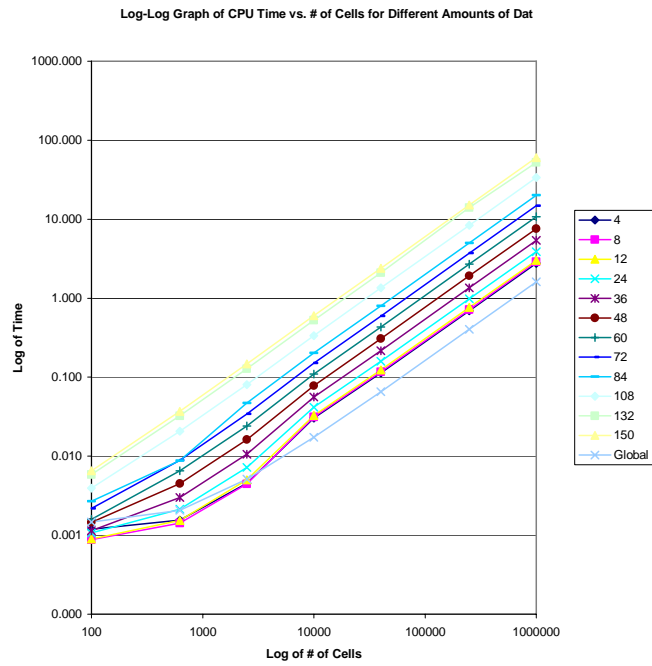


Figure 3: Log-Log graph of CPU time versus number of cells for different amounts of data

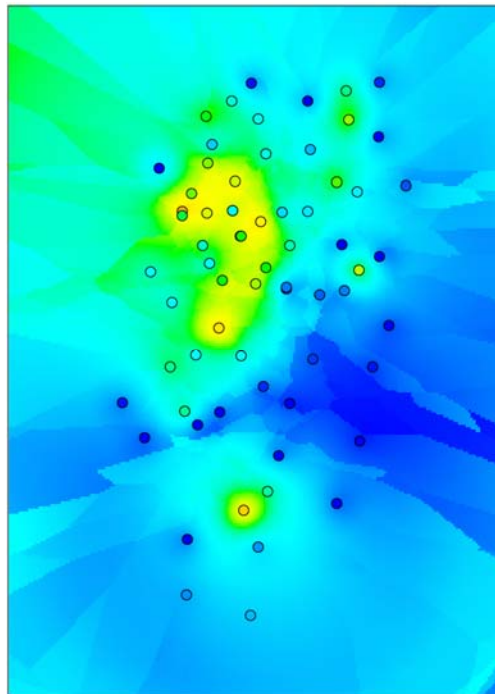


Figure 4: Ordinary kriging with a maximum of 8 samples.

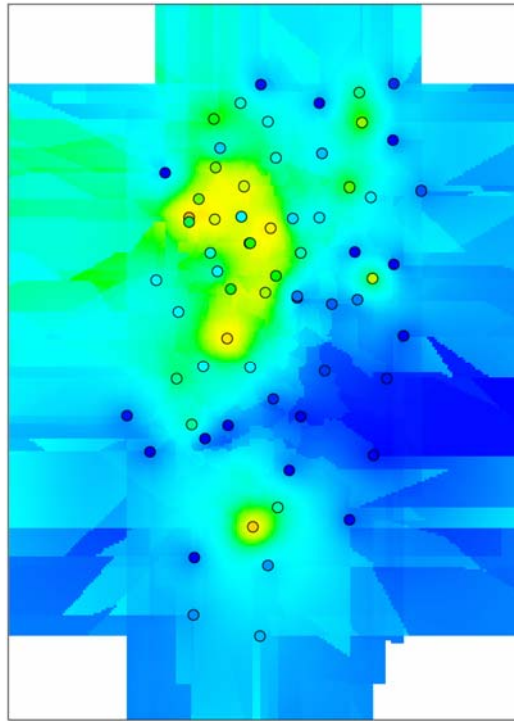


Figure 5: Ordinary kriging with a maximum of 8 samples and an octant search.

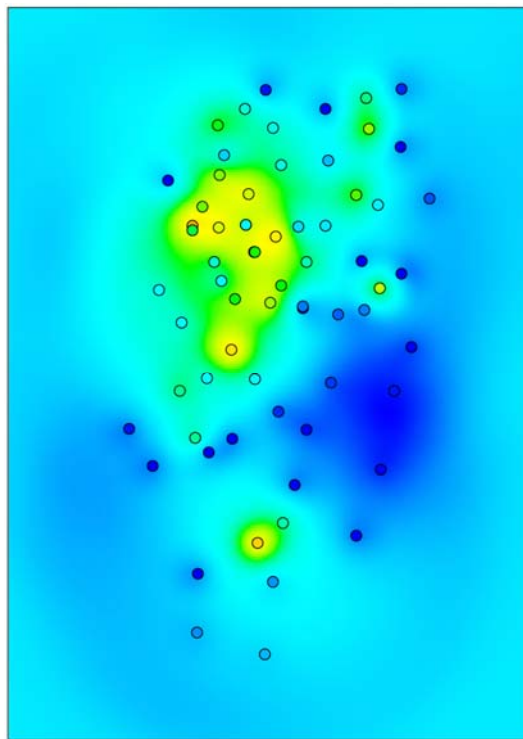


Figure 6: Global kriging using all of the sample data.