A Global Kriging Program for Multiple Variables

Brandon J. Wilde and Clayton V. Deutsch

Global kriging is a useful method for generating artifact-free maps. The local search is eliminated as the left-hand side of the system of equations is constant for all estimation locations. Since it is constant, this potentially large square matrix must only be decomposed once. This can lead to a large reduction in computation time. Often, geologic variables are estimated accounting for their own spatial relationship as well as the spatial relationship with other variables. A program for performing global kriging with the ability to account for multiple variables is presented. The theory and implementation are discussed. The ability to consider a data-error coordinate is also implemented. The computer expenses, memory and time, are considered. Both have a quadratic relationship with the number of data.

Introduction

Kriging is a spatial interpolation method used in the estimation of geologic variables such as mineral grades, petrophysical properties, and contaminant concentrations. 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 calculates an estimate, $Z^*(\mathbf{u})$, at a location by assigning weights, λ_i, to a number, n, of nearby data and using the data values, $Z(\mathbf{u}_i)$, to arrive at the weighted average shown in Equation 1 . The weights assigned to each data point are determined by solving a system of equations consisting of the covariances between data points i and j, C_{ij}, and the covariances between the data points and an unsampled location, C_{i0}, where $i,j=1,\ldots,n$ as shown in Equation 2. The computational speed of kriging is controlled by the size of the system of equations which is controlled by the number, n, of nearby data used for calculating an estimate; choosing to use a smaller number of data results in faster computation. Unfortunately, using a small number of data to calculate estimates can result in maps of estimates with unrealistic artifacts. To avoid the creation of these artifacts, the practitioner increases the number of data used to calculate estimates. Even when many data are used, there may still be artifacts in the maps.

$$Z^*(\mathbf{u}) - m(\mathbf{u}) = \sum_{i=1}^{n} \lambda_i \cdot \left[Z(\mathbf{u}_i) - m(\mathbf{u}_i) \right] \quad (1)$$

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

Neufeld and Wilde (2005) implemented a kriging program which uses all of the data to estimate all locations. This approach, termed global kriging, eliminates artifacts and can be faster than the traditional implementation of kriging. The improved computational efficiency comes from the recognition that the left-hand side (LHS) matrix of C_{ij} values remains constant for all estimation locations. Let

$$[K] = \begin{bmatrix} C_{11} & \cdots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \cdots & C_{nn} \end{bmatrix}, [k] = \begin{bmatrix} C_{10} \\ \vdots \\ C_{n0} \end{bmatrix}, [\lambda] = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}, \text{and } [z_\alpha] = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix} \quad (3)$$

Equation 2 can be re-expressed as

$$[K][\lambda] = [k] \quad (4)$$

The Cholesky decomposition of K yields $K = R^T R$ such that R is upper triangular and has all main diagonal entries positive. Substituting into Equation 4 yields $R^T R\lambda = k$. Let $y = R\lambda$. λ is unknown and therefore y is unknown also. However, y satisfies $R^T y = k$. Since R^T is lower triangular, y can be determined by forward substitution (Cormen *et*

al., 2009). Once *y* is determined, the upper triangular system $R\lambda = y$ is used to solve for λ by back substitution (Watkins, 2002).

For global kriging, the *K* matrix is constant for all estimation locations. Thus, the step of decomposing the *K* matrix to get *R* needs only be performed once. Only the *k* and λ vectors are updated for each estimation location to calculate $Z^*(\mathbf{u})$. This allows the calculation of artifact-free maps to be performed relatively quickly.

The current implementation of global kriging (`kt3d_gl`) allows the user to perform simple kriging, ordinary kriging, kriging with a locally varying mean (LVM), and kriging with external drift (KED). This work documents a global kriging program that is able to:

- Cokrige multiple variables simultaneously
- Perform collocated cokriging with multiple secondary variables
- Perform kriging on non-isotopic data using a fourth dimension

This program will also perform lower-level global kriging operations such as simple global kriging of one variable.

Because of differences in units of measurement, the variances of the different variables may differ by several orders of magnitude. This leads to large differences between rows of the cokriging matrix. This may cause numerical instability when solving the cokriging system. As such, a modification to the general kriging algorithm is made to ensure the stability of the system of equations. Each variable is standardized to have a mean of zero and a variance of one. The user inputs the variable means and variances and the following relation is used to standardize the values:

$$y_{i,j} = \frac{z_{i,j} - m_j}{\sigma_j}$$

where m_j is the mean of the j^{th} variable, σ_j is the standard deviation of the j^{th} variable, $z_{i,j}$ is the i^{th} sample of the j^{th} variable, and $y_{i,j}$ is the i^{th} standardized sample of the j^{th} variable. The sills of all direct variograms are 1.0. An error occurs if this is not the case. This standardization step ensures that the covariances which populate the kriging system will be of similar magnitude causing the kriging system to enjoy increased stability. Estimates are calculated in this standardized space and the standardized estimates are de-standardized by rearranging the above equation to solve for *z*.

Global Cokriging

Consider the situation where primary data $\left\{z_1(\mathbf{u}_{\alpha_1}), \alpha_1 = 1,\ldots,n_1\right\}$ are supplemented by $N_v - 1$ secondary continuous attributes z_i, $\left\{z_i(\mathbf{u}_{\alpha_i}), \alpha_i = 1,\ldots,n_i, i = 2,\ldots,N_v\right\}$. The kriging estimator (Equation 1) can be extended to incorporate that additional information:

$$Z_1^*(\mathbf{u}) - m_1(\mathbf{u}) = \sum_{i_1=1}^{n_1} \lambda_{i_1} \cdot \left[Z(\mathbf{u}_{i_1}) - m(\mathbf{u}_{i_1})\right] + \sum_{\alpha=2}^{N_v} \sum_{i_\alpha=1}^{n_\alpha} \lambda_{i_\alpha} \cdot \left[Z_\alpha(\mathbf{u}_{i_\alpha}) - m(\mathbf{u}_{i_\alpha})\right] \quad 5$$

where λ_{i_1} are the weights assigned to the primary data and λ_{i_α} are the weights assigned to the $N_v - 1$ secondary data. Note that Equation 5 relates to global kriging in that the $n_\alpha, \alpha = 1,\ldots,N_v$ do not depend on **u**, that is, all of the data for all N_v data types are used. This has the potential to make the *K* matrix become very large. Consider that $N_v = 2$ and that $n_1 = n_2$. The global kriging *K* matrix will have four times as many elements than if only the primary data were used.

To solve for the weights in Equation 5, the following system must be solved:

$$\begin{bmatrix} \left[C_{11}\left(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_1}\right)\right] & \cdots & \left[C_{1N_v}\left(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_{N_v}}\right)\right] \\ \vdots & \ddots & \vdots \\ \left[C_{N_v1}\left(\mathbf{u}_{\alpha_{N_v}}-\mathbf{u}_{\beta_1}\right)\right] & \cdots & \left[C_{N_vN_v}\left(\mathbf{u}_{\alpha_{N_v}}-\mathbf{u}_{\beta_{N_v}}\right)\right] \end{bmatrix} \begin{bmatrix} \left[\lambda_{\beta_1}(\mathbf{u})\right] \\ \vdots \\ \left[\lambda_{\beta_{N_v}}(\mathbf{u})\right] \end{bmatrix} = \begin{bmatrix} \left[C_{11}\left(\mathbf{u}_{\alpha_1}-\mathbf{u}\right)\right] \\ \vdots \\ \left[C_{N_v1}\left(\mathbf{u}_{\alpha_{N_v}}-\mathbf{u}\right)\right] \end{bmatrix} \quad 6$$

where $\left[C_{ij}\left(\mathbf{u}_{\alpha_i}-\mathbf{u}_{\beta_j}\right)\right]$ is the $n_i \times n_j$ matrix of data-data direct and cross covariances, $\left[\lambda_{\beta_i}(\mathbf{u})\right]$ is a vector of cokriging weights, and $\left[C_{i1}\left(\mathbf{u}_{\alpha_i}-\mathbf{u}\right)\right]$ is a vector of data-to-unknown direct and cross covariances. The cokriging weights are obtained by decomposing the LHS matrix and performing forward and back substitution with the resulting triangular *R* matrix. The LHS matrix can be very large when there is a large number of data or when there are many secondary variables.

An example of the system of equations for three variables is shown in **Figure 1**. There are $n = n_1+n_2+n_3$ samples leading to a LHS of size *n* x *n*. The sub-matrices and subvectors shaded in red are covariances calculated from direct variograms. The sub-matrices and subvectors shaded in blue are covariances calculated from cross variograms. The size of the system of equations can become very large when considering multiple variables.

The global kriging program documented herein is able to perform global cokriging. The direct and cross variograms are specified the same as for the GSLIB program cokb3d (Deutsch & Journel, 1998). A global cokriging example comes from the data used in Problem Set Four in Deutsch & Journel (1998). There are 29 primary and 2500 secondary data within the 50 x 50m area. The 29 primary values and the 29 collocated secondary values (**Figure 2**) are used to estimate at the 2500 locations. The direct and cross variograms come from Deutsch & Journel (1998). The LHS matrix consists of the $(29 + 29)^2 = 3364$ direct and cross covariances. This matrix is decomposed and used to determine the weights at each of the 2500 locations. The estimates are shown in **Figure 3**. They are similar to those shown in Deutsch & Journel with the absence of any search artifacts.

The cokriging theory can be extended to the estimation of multiple variables simultaneously. The LHS matrix remains the same while the RHS vector is updated for the estimation of each variable at a location. This is illustrated in **Figure 4**. The covariance values shaded in red are calculated from direct variograms while those shaded in blue come from cross variograms. The number of variables to be estimated is specified in the parameter file as the number of *primary* variables.

A global kriging system with three variables requires three direct variograms and three cross variograms. The inference of these variograms becomes demanding in terms of data as the number of variables increases. As well, the joint modeling of these variograms is particularly tedious. As such, cokriging has not been extensively used in practice (Deutsch & Journel, 1998).

Collocated Cokriging

Collocated cokriging is a modification of cokriging arising from the recognition that secondary data can be more densely sampled than primary data and consists of retaining only the collocated secondary variables $z_\alpha(\mathbf{u}), \alpha = 1,...,N_v$, provided they are available at all locations being estimated. In a global kriging context, it is not feasible to consider all the exhaustive secondary data; the kriging system would be much too large. The simplification of considering only those secondary data collocated with the location being estimated avoids this problem. This is no longer global kriging in its pure definition; not all of the data are used to estimate every location. The LHS of the system of equations will vary from location to location as different data are used at each location. Fortunately, the speed advantage of global kriging can be maintained even when parts of the LHS change from location to location, provided these changes affect only a small portion of the LHS matrix.

Watkins (2002) describes a technique for decomposing a matrix by blocks. Consider a primary variable with n_1 samples and $N_v - 1$ secondary data available exhaustively. The system of equations is composed as

shown in **Figure 5** where *i* and *j*=2,…,N_v-1. The LHS matrix has size *n* x *n* where $n = n_1 + N_v - 1$. The sub-matrix shaded in red is the n_1 x n_1 matrix of primary data-data covariances. The blue shaded sub-matrices are of size n_1 x (N_v-1) and (N_v-1) x n_1 and contain the covariances between the n_1 data and the N_v-1 collocated secondary variables. These covariances are often approximated as: $C_{i1}(\mathbf{u}_{\alpha_i} - \mathbf{u}) = B \cdot C_{11}(\mathbf{u}_{\alpha_i} - \mathbf{u})$ where $B = \sqrt{C_i(0)/C_1(0)} \cdot \rho_{i1}(0)$, $C_1(0)$, $C_i(0)$ are the variances of Z_1 and Z_i, and $\rho_{i1}(0)$ is the linear correlation coefficient of collocated $z_1 - z_i$ data, *i*=2,…,N_v-1 (Deutsch & Journel, 1998). The green shaded sub-matrix is comprised of the correlation coefficients between the N_v-1 secondary data. The purple shaded subvector is the vector of primary data-unsampled location covariances and the orange shaded subvector is the vector of correlation coefficients between the primary variable and the N_v-1 secondary variables.

To implement the Cholesky decomposition by blocks, let $K_{11} = C_{11}$, $B = C_{1j}$ ($B^T = C_{i1}$), and $\hat{K} = C_{ij}$, $i,j > 1$. The Cholesky Decomposition Theorem states that for positive definite *K*, *K* can be decomposed in exactly one way into a product $K = R^T R$ such that *R* is upper triangular and has all main diagonal entries positive. *R* is called the Cholesky factor of *K* (Watkins, 2002). A block form of the product $K = R^T R$ is shown in Equation 8. K_{11} and R_{11} are square matrices. K_{11} is symmetric and positive definite. Equating the blocks gives equations 9, 10, and 11.

$$\begin{bmatrix} K_{11} & B \\ B^T & \hat{K} \end{bmatrix} = \begin{bmatrix} R_{11}^T & 0 \\ S^T & \hat{R}^T \end{bmatrix} \begin{bmatrix} R_{11} & S \\ 0 & \hat{R} \end{bmatrix} \quad (8)$$

$$K_{11} = R_{11}^T R_{11} \quad (9)$$

$$B = R_{11}^T S \quad (10)$$

$$\hat{K} = S^T S + \hat{R}^T \hat{R} \quad (11)$$

R_{11}, *S,* and $\hat{R}$ are determined as follows:

$$R_{11} = cholesky(K_{11})$$
$$S = R_{11}^{-T} B$$
$$\tilde{K} = \hat{K} - S^T S$$
$$\hat{R} = cholesky(\tilde{K})$$

where R_{11}^{-T} means $\left(R_{11}^{-1}\right)^T$. K_{11} is the matrix of data-data covariances and remains the same for all locations, therefore R_{11} also remains the same for all locations. Only *B* varies from location to location. *S* is updated at each location and used to determine $\tilde{K}$ whose Cholesky decomposition gives $\hat{R}$ (Watkins, 2002). The upper triangular matrix composed of R_{11}, *S*, and $\hat{R}$ is used to solve for the vector of kriging weights, λ. As *Nv* is typically small, the updating of the *R* matrix at each location proceeds quickly.

Collocated cokriging as described is implemented in the global kriging program documented herein. The number of covariates is specified in the parameter file. For each covariate, the name of the file containing the covariate is specified as well as the column number, mean, and variance of the covariate. The correlations between covariates are required as well as the correlations between the covariates and the primary variable(s). The name of the file and the column number containing the correlation matrix with these correlations is specified in the parameter file. The correlation matrix has size *Nv* x *Nv*.

The data in Problem Set Four from Deutsch & Journel is used to show an example of global collocated cokriging. There are 29 primary and 2500 secondary data within a 50 x 50m area (**Figure 6**). At each of the estimation locations, the 29 primary data and the single collocated secondary data are used for estimation. The bulk of the LHS matrix remains the same as it is comprised of the data-data covariances between the 29 data. The final row and column of the LHS matrix is updated at each estimation location. The block Cholesky method is used

to update the decomposition of the LHS matrix to get the weights. The estimates calculated using this method are shown in **Figure 7**. They are similar to those in the example provided by Deutsch & Journel (1998).

Data Error Coordinate

Different data sources have different errors. Deutsch *et al.* (2010) proposed a modification to kriging to account for the varying quality of data. It amounts to adding an additional coordinate to each data. The magnitude of this coordinate corresponds to the error in the data. This additional coordinate is incorporated into the calculation of the distance between two points, whether they are two data locations or a data location and an unsampled location. This additional coordinate is an option in the global kriging program documented herein. The capability exists for each of the primary variables to have a unique error coordinate. This is specified as a column number; a column number less than 1 means no error coordinate is considered.

String Effect

The string effect is a well-known phenomenon that occurs when performing kriging with data contiguously aligned along finite strings (Deutsch, 1994). The samples at the end of such strings are seen by the kriging system as being less redundant than the other samples in the string. As such, more weight is given to the samples at the ends of the string than is given to the other samples in the string. Global kriging is not exempt from this effect. This is tested by estimating on a 2D plane with a string of data down the center as shown in **Figure 8**. The cumulative sum of the kriging weights assigned to each data is shown by the gray line in **Figure 8**. The samples at the ends of the string received the greatest weight while the samples adjacent to the end samples received the least weight. The weight then increases towards the center of the string.

Expenses: Memory and Time

Using all of the data to estimate at every location can lead to very large systems of equations. Computers have finite memory capacity for storing the values which comprise these systems. Double precision floating point values require 64 bits of memory. In one gigabyte of RAM, there are more than 8.5 billion bits, enough memory to store more than 100 million double precision floating point values. Recall that when there are n data, the system actually requires $n^2 + 2n$ entries in the memory (n^2 for the LHS matrix, n for the weights vector, and n for the RHS vector). Therefore, with one gigabyte of memory, a system with n= 10,000 data can be stored (entries in memory =100,020,000). The memory required increases quadratically with the number of data; n = 20,000 would require approximately 4GB of memory. The relationship between the number of data and RAM required is shown in **Figure 9**. 32 bit systems are only capable of utilizing 4 GB of RAM; utilizing more RAM than this requires a 64 bit system.

The computational time required to evaluate the systems can be prohibitive. A system with n =10,000 data requires the calculation of more than 50 million covariances. That is just to build the system. The system must then be solved. The Cholesky factorization of a real symmetric positive definite matrix as implemented in the global kriging program discussed requires $O(n^3)$ computations to decompose the LHS matrix. This is more than 333 billion computations for n = 10,000. Once the matrix is decomposed, the resulting triangular matrix is used to solve for the weights at each location. This step requires a further $O(n^2)$ computations at each location. There are typically many more locations to estimate than there are data. The computation time for global kriging is dominated by the forward and back substitution steps to solve for the weights. The time increases linearly with the number of estimation locations and quadratically with the number of data.

Time Comparison with Traditional Search-based Kriging

It is interesting to consider the amount of time required to perform global kriging and to relate this to the time it takes to perform traditional search-based kriging. To compare, a grid with 100,000 cells is estimated using both

global and search-based kriging. Three data sets with 250, 500, and 750 data are used. The number of data used to estimate for search-based kriging is varied from 10 to 240. The results of this time comparison are shown in **Figure 10**. This figure compares run time with the number of data used in the search. Of course, global kriging does not use a search and the run time is constant; it depends only on the total number of data and the number of estimation locations. The run time for search-based kriging is highly dependent on the number of data used to estimate at each location. Using many data leads to a large system of equations to be solved at every location increasing the run time for the estimation. The total number of data has a slight impact on the run time for search based kriging; estimating takes longer when there are 750 data than when there are 500 data which in turn takes longer than where there are 250 data. The run times of global kriging and search-based kriging are equivalent when the search is set to use approximately 10% of the total number of data. When estimation was performed using 250 data, the time to perform global kriging was 8 seconds. Search-based kriging took approximately 8 seconds when 25 data were used to estimate. When 750 data were used, global kriging took 39 seconds. Search-based kriging took approximately 39 seconds when 75 data were used. As a rule-of-thumb, search-based kriging is faster when fewer than 10% of the data are used to estimate at each location; when more data than this are used, global kriging is faster.

Conclusions

Global kriging is effective for producing artifact-free maps. The lack of artifacts is achieved by using all the data to estimate at all locations. Using all the data to estimate all locations would be extremely computer intensive as the large system of equations would have to be solved at every location. Global kriging capitalizes on the fact that the LHS matrix is constant for all estimation locations. This matrix must therefore only be decomposed once. The decomposed matrix can then be used at all estimation locations to determine the vector of weights used to calculate the estimates. There was previously no capability for performing global cokriging or global collocated cokriging. The software is available for performing these tasks. Global cokriging is straightforward; and LMC is input and all of the data regardless of type are used to estimate at all locations. Global collocated cokriging is easier to implement as the LMC is not required, but some important changes to the algorithm for systems solving is required. The LHS matrix is decomposed by blocks to accommodate the changes to the matrix at each new estimation location. The capability for considering an additional data error coordinate is also incorporated. The main computer resource considerations for global kriging are memory and time. Memory has a quadratic relationship with the number of data; 10,000 data require approximately 1GB of RAM, 20,000 data require approximately 4GB of RAM. The computation time is dominated by the number of estimation locations, the time to decompose the LHS matrix is not significant.

References

Cormen TH, Leiserson CE, Rivest RL, and Stein C, 2009. Introduction to Algorithms. MIT Press, Cambridge. 1292p.

Deutsch CV, 1994. Kriging with strings of data. Mathematical Geology, Vol. 26, No. 5, p623-638

Deutsch CV and Journel AG, 1998. GSLIB: Geostatistical Software Library and User's Guide 2nd edition. Oxford University Press, New York. 369p.

Deutsch JL, Boisvert JB, and Deutsch CV, 2010. A new dimension to account for data error and volume support. Centre for Computational Geostatistics: Report 12. University of Alberta. p308-1 to 308-16.

Goovaerts P, 1997. Geostatistics for Natural Resources Evaluation. Oxford University Press, New York. 483p.

Neufeld CT and Wilde BJ, 2005. A global kriging program for artifact-free maps. Centre for Computational Geostatistics: Report 7. University of Alberta. P403-1 to 401-8.

Watkins DJ, 2002. Fundamentals of Matrix Computations. Wiley-Interscience, New York. 618p.

Figures

$$\begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_1}) & C_{12}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_2}) & C_{13}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_3}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_1}) & C_{22}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_2}) & C_{23}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_3}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_1}) & C_{32}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_2}) & C_{33}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_3}) \end{bmatrix} \bullet \begin{bmatrix} \lambda_{\beta_1}(\mathbf{u}) \\ \lambda_{\beta_2}(\mathbf{u}) \\ \lambda_{\beta_3}(\mathbf{u}) \end{bmatrix} = \begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}) \end{bmatrix}$$

Figure 1: System of equations for cokriging with three variables.

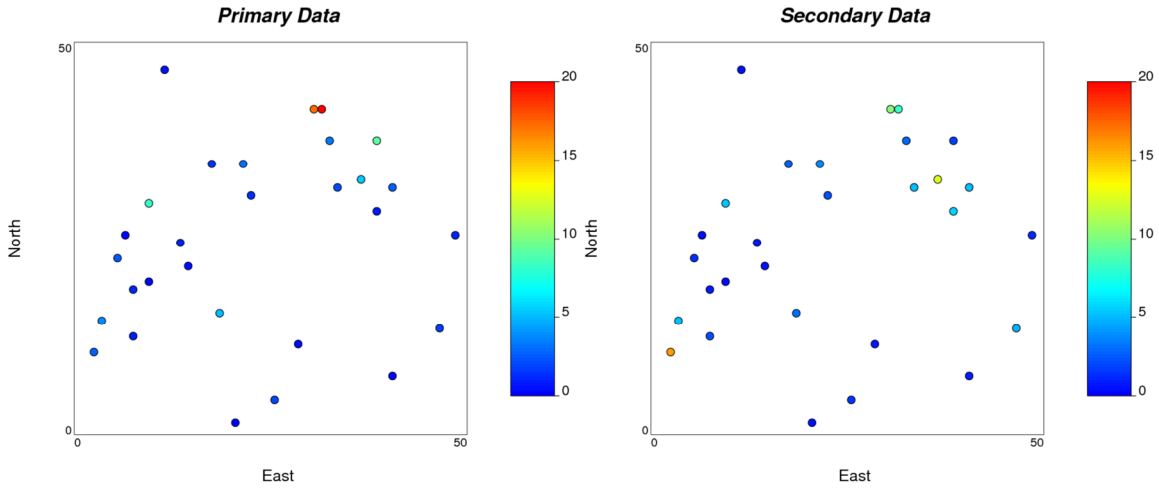

Figure 2: Primary and secondary data for the global cokriging example.

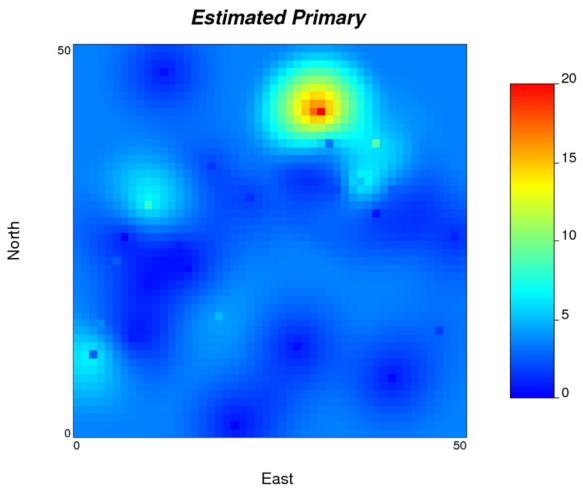

Figure 3: Primary variable estimated by global cokriging.

$$\begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_1}) & C_{12}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_2}) & C_{13}(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_3}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_1}) & C_{22}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_2}) & C_{23}(\mathbf{u}_{\alpha_2}-\mathbf{u}_{\beta_3}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_1}) & C_{32}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_2}) & C_{33}(\mathbf{u}_{\alpha_3}-\mathbf{u}_{\beta_3}) \end{bmatrix} \bullet \begin{bmatrix} \lambda_{\beta_1}(\mathbf{u}) \\ \lambda_{\beta_2}(\mathbf{u}) \\ \lambda_{\beta_3}(\mathbf{u}) \end{bmatrix} = \begin{matrix} \text{Estimating } z_1 \\ \begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}) \end{bmatrix} \end{matrix} , \begin{matrix} \text{Estimating } z_2 \\ \begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}) \end{bmatrix} \end{matrix} , \begin{matrix} \text{Estimating } z_3 \\ \begin{bmatrix} C_{11}(\mathbf{u}_{\alpha_1}-\mathbf{u}) \\ C_{21}(\mathbf{u}_{\alpha_2}-\mathbf{u}) \\ C_{31}(\mathbf{u}_{\alpha_3}-\mathbf{u}) \end{bmatrix} \end{matrix}$$

Figure 4: System of equations for the simultaneous estimation of multiple variables.

$$\begin{bmatrix} C_{11}\left(\mathbf{u}_{\alpha_1}-\mathbf{u}_{\beta_1}\right) & C_{1j}\left(\mathbf{u}-\mathbf{u}_{\beta_j}\right) \\ C_{i1}\left(\mathbf{u}_{\alpha_i}-\mathbf{u}\right) & C_{ij}\left(\mathbf{u}-\mathbf{u}\right) \end{bmatrix} \bullet \begin{bmatrix} \lambda_{\beta_1}(\mathbf{u}) \\ \lambda_{\beta_i}(\mathbf{u}) \end{bmatrix} = \begin{bmatrix} C_{11}\left(\mathbf{u}_{\alpha_1}-\mathbf{u}\right) \\ C_{i1}\left(\mathbf{u}-\mathbf{u}\right) \end{bmatrix}$$

Figure 5: System of equations for collocated cokriging.

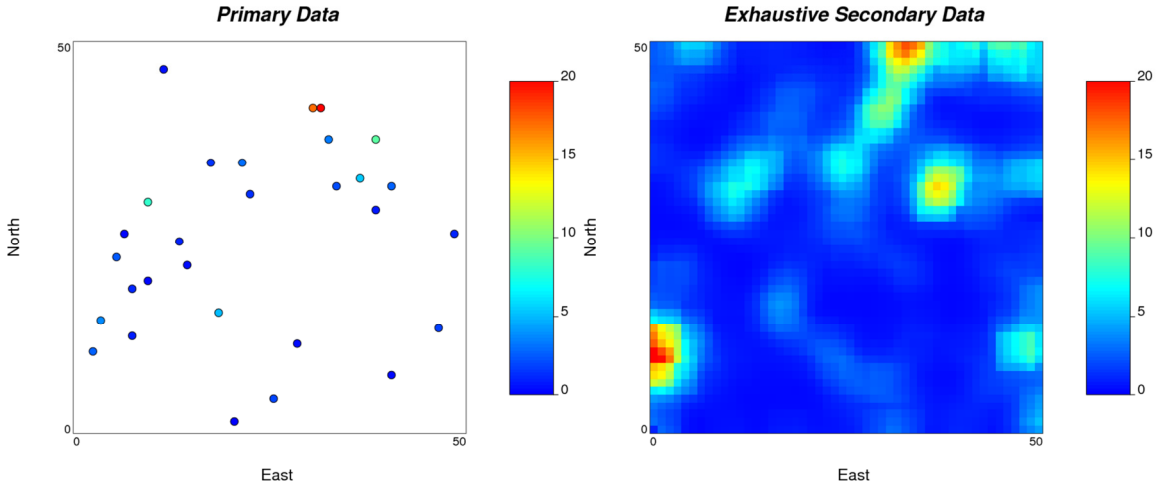

Figure 6: Primary and secondary data for global collocated cokriging example.

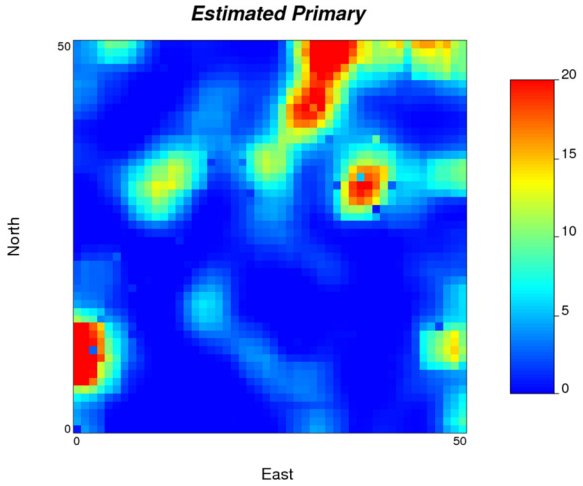

Figure 7: Estimates calculated using global collocated cokriging.

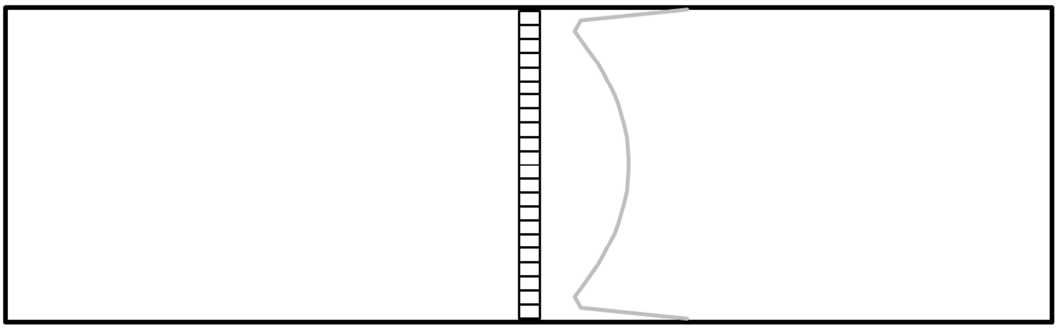

Figure 8: The weights assigned to a string of data by global kriging.

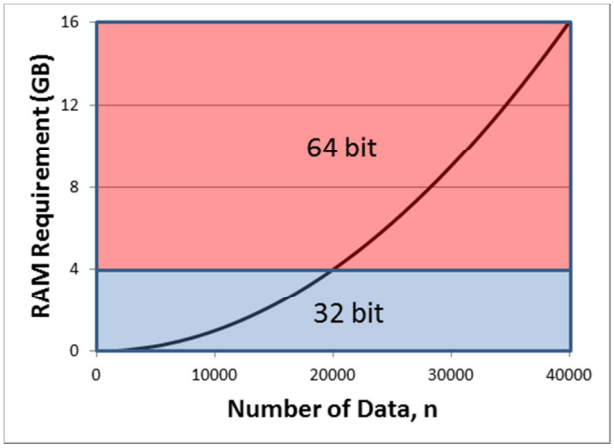

Figure 9: RAM requirement vs. number of data for global kriging.

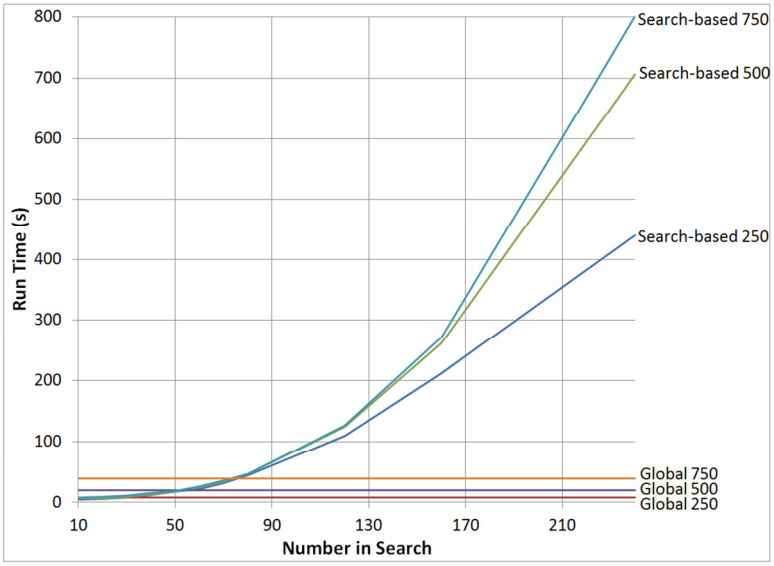

Figure 10: Run time comparison for global and search-based kriging.