

# A Fast Algorithm to Enforce Nonnegative Kriging Weights or Nonnegative Kriging Estimates

Behrang Koushavand and Clayton V. Deutsch

Negative weights in Kriging arise when close data in the neighborhood of estimation screen outlying data. The negative weights can be significant depending on the variogram and the data configuration. Negative weights may be desirable when estimating smooth variables and continuous phenomena such as surfaces or thickness values; they permit extrapolation of local trends. Negative weights, however, cause Kriging to be a nonconvex estimator, that is, the estimate may be less than the minimum data value or greater than the maximum data value. This is a good feature when the estimation is accounting for a local trend in the data values; however nonconvex estimation may cause negative or nonphysical estimates. Mass or volume fractions must be between 0 and 1. Probabilities must also be between 0 and 1. Rate constants must be non negative. Previous authors have presented nonlinear programming algorithms to minimize the estimation variance under a constraint of non-negative Kriging weights (Barnes and Johnson, 1984; Szidarovszky, Baafi and Kim, 1987; Herzfeld, 1989). These methods are relatively CPU intensive and complex.

The approach presented here is based on Rosen's gradient method, which gives the same solution with much less CPU time and with a simpler formulism. This algorithm can be used for any linear or nonlinear type of Kriging including: Simple Kriging (SK), Ordinary Kriging (OK) and Universal Kriging (UK).

## Simple Kriging (SK)

Consider the estimate of an unsampled value  $z(u)$  from neighboring data values  $z(u_\alpha); \alpha = 1, 2, \dots, n$ . The Simple Kriging (SK) estimator is written as:

$$z_{SK}^*(u) = \sum_{\alpha=1}^n \lambda_\alpha(u) \cdot [z(u_\alpha) - m] + m \quad (1)$$

Where  $m$  is the mean of regionalized variable. The simple Kriging (SK) estimator is unbiased since the error mean is equal to zero, (Goovaerts, 1997). The Kriging Variance is calculated as follow:

$$\sigma_E^2(u) = C(0) - 2 \sum_{\alpha=1}^n \lambda_\alpha \cdot C_{u\alpha} + \sum_{\alpha=1}^n \sum_{\beta=1}^n \lambda_\alpha \cdot \lambda_\beta \cdot C_{\alpha\beta} \quad (2)$$

The model of nonlinear programming for Simple Kriging can be written as:

$$\text{Min } f(\lambda_1, \dots, \lambda_n) = f(\mathbf{x}) = \sigma_E^2(u) \quad (3)$$

The gradient of function  $f$  is written as:

$$\mathbf{g}(\mathbf{x}) = \nabla f(\lambda) = \begin{bmatrix} \frac{\partial f}{\partial \lambda_1} \\ \vdots \\ \frac{\partial f}{\partial \lambda_n} \end{bmatrix}_{n \times 1}$$

The gradient also can be written as:

$$\mathbf{g}(\mathbf{x}) = 2 \cdot \mathbf{C} \cdot \mathbf{x} - 2 \cdot \mathbf{r} \quad (4)$$

Where:

$$\mathbf{C} = \begin{bmatrix} \bar{C}_{11} & \cdots & \bar{C}_{n1} \\ \vdots & \ddots & \vdots \\ \bar{C}_{1n} & \cdots & \bar{C}_{nn} \end{bmatrix}_{n \times n} \quad \mathbf{x} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}_{n \times 1} \quad \mathbf{r} = \begin{bmatrix} \bar{C}_{u1} \\ \vdots \\ \bar{C}_{un} \end{bmatrix}_{n \times 1}$$

We can add any linear equality or non equality constrains to this model.

$$\begin{aligned} & \text{Minimize } f(\mathbf{x}) \\ & \text{Subject to } \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{E}\mathbf{x} = \mathbf{e} \end{aligned} \quad (5)$$

### SK With Nonnegative Weights

The constraint of non-negative Simple Kriging weights is:

$$-\lambda_{\alpha}(\mathbf{u}) \leq 0 \quad \forall \alpha = 1..n$$

A is a diagonal matrix with diagonal elements equal to -1 as follows:

$$\mathbf{A} = \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}_{n \times n} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}_{n \times 1} \quad (6)$$

### SK With Constrained Estimation Values

If we want to set  $Z_L < Z_{SK}^*(\mathbf{u}) < Z_U$  then

$$\mathbf{A} = \begin{bmatrix} -z(u_1) & \cdots & -z(u_n) & z_L \\ z(u_1) & \cdots & z(u_n) & -z_U \end{bmatrix}_{2 \times n}, \quad \mathbf{x} = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_n \end{bmatrix}_{n \times 1}, \quad \mathbf{E} \text{ and } \mathbf{e} \text{ are empty.}$$

### Ordinary Kriging (OK)

Ordinary Kriging (OK) is like simple kriging with the linear constraint that the weights sum to one:

$$\sum_{\alpha=1}^n \lambda_{\alpha}(\mathbf{u}) = 1$$

So:  $\mathbf{E} = [1 \quad \dots \quad 1]_{1 \times n}$  and  $\mathbf{e} = [1]_{1 \times 1}$ .

Non-negative kriging weights or constrained estimate value can be applied to OK system the same as SK.

### Universal Kriging (UK)

Consider the linear trend model in 2D:  $m(x, y) = a_0 + a_x x + a_y y$ . So E and e are defined as:

$$\mathbf{E} = \begin{bmatrix} 1 & \cdots & 1 \\ x_1 & \cdots & x_n \\ y_1 & \cdots & y_n \end{bmatrix}_{3 \times n}, \quad \mathbf{e} = \begin{bmatrix} 1 \\ x_0 \\ y_0 \end{bmatrix}_{3 \times 1}$$

According to the trend model,  $\mathbf{E}$  and  $\mathbf{e}$  can be easily defined.

### Rosen's Gradient Projection Algorithm

Kriging variance is a function of  $\lambda_i; i = 1..n$  that should be minimized subject to some constrains that defined above. Barnes and Johnson (1984) and Szidarovszky (et. al. 1987) defined a new function such that all constrains are honed in this new function. To optimization this function, they suggested numerical algorithms that are based on random path searching algorithms and very CPU intensive.

The negative value of a gradient vector of a function is direction of maximum decrees of that function. This property is used in all gradient based optimization methods. In these algorithms, negative of gradient vector is used to improve initial answer and new point calculated; gradient of function should be recalculated at all new points. These algorithms are much faster than searching methods.

To use constraints with gradient methods, Rosen (1960) used projection of gradient vector on the each of constrains. So this new direction always honor constrains and also is a direction to minimize the function.

Stages of algorithm are as following (Rosen 1960):

Initialization Step: Chose an initial answer of  $\mathbf{x}_1$  with  $\mathbf{A}\mathbf{x}_1 \leq b$  and  $\mathbf{E}\mathbf{x}_1 = e$ . Suppose that  $\mathbf{A}^t$  and  $b^t$  are decomposed into  $(\mathbf{A}_1^t, \mathbf{A}_2^t)$  and  $(\mathbf{b}_1^t, \mathbf{b}_2^t)$  such that  $\mathbf{A}_1\mathbf{x}_1 = \mathbf{b}_1$  and  $\mathbf{A}_2\mathbf{x}_1 < \mathbf{b}_2$ . Let  $k=1$  and go to the main step.

Main step

1. Let  $\mathbf{M}^t = (\mathbf{A}_1^t, \mathbf{E}^t)$ . If  $\mathbf{M}$  is empty, let  $\mathbf{P}=\mathbf{I}$  (identical matrix); otherwise, let  $\mathbf{P} = \mathbf{I} - \mathbf{M}^t(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}$ . Let  $\mathbf{d}_k = -\mathbf{P}\mathbf{g}(\mathbf{x}_k)$ . If  $\mathbf{d}_k \neq 0$  go to step 2. If  $\mathbf{d}_k = 0$ , stop if  $\mathbf{M}$  is empty; otherwise, let  $w = -(\mathbf{M}\mathbf{M}^t)^{-1}\mathbf{M}\mathbf{g}(\mathbf{x}_k)$  and let  $\mathbf{w}^t = (\mathbf{u}^t, \mathbf{v}^t)$ . If  $\mathbf{u} \geq 0$ , Stop;  $\mathbf{x}_k$  is the Kuhn-Tucker point. Else chose a negative component of  $\mathbf{u}$ , say  $\mathbf{u}_j$ . Update  $\mathbf{A}_1$  by deleting the row corresponding to  $\mathbf{u}_j$ , and repeat step 1.

2. Calculate  $\alpha$  by following equations:

$$\alpha_1 = \frac{\mathbf{d}_k^t \cdot \mathbf{b} - \mathbf{x}_k^t \cdot \mathbf{C} \cdot \mathbf{d}_k}{\mathbf{d}_k^t \cdot \mathbf{C} \cdot \mathbf{d}_k}$$

$$\hat{\mathbf{b}} = \mathbf{b}_2 - \mathbf{A}_2 \cdot \mathbf{x}_k$$

$$\hat{\mathbf{d}} = \mathbf{A}_2 \cdot \mathbf{d}_k$$

$$\alpha_2 = \text{Min} \left| \frac{\hat{\mathbf{b}}_j}{\hat{\mathbf{d}}_j} \right|; \hat{\mathbf{d}}_j > 0$$

$$\alpha = \text{Min} \{ \alpha_1, \alpha_2 \}$$

Let  $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \cdot \mathbf{d}_k$ . Replace  $k$  by  $k+1$  and suppose that  $\mathbf{A}^t$  and  $\mathbf{b}^t$  are decomposed into  $(\mathbf{A}_1^t, \mathbf{A}_2^t)$  and  $(\mathbf{b}_1^t, \mathbf{b}_2^t)$  such that  $\mathbf{A}_1\mathbf{x}_k = \mathbf{b}_1$  and  $\mathbf{A}_2\mathbf{x}_k < \mathbf{b}_2$ . Repeat step 1.

The `rosen` program has been provide based on GSLIB (Deutsch and Journal 1998) which solves Kriging system with Rosen gradient method.

The parameter file of `Rosen` program is the same as `kt3d` (Deutsch and Jurnel 1998) but has extra parameters at the last line. `Stype` is an integer value as bellow:

- 0: Use Lagrange multiplier to solve the kriging system; identical to `kt3d`.
- 1: Use Rosen method without any constraints.
- 2: Use Rosen with positive weights.
- 3: Use Rosen with  $Z^* \geq z \text{ min}$
- 4: Use Rosen with  $Z^* \leq z \text{ max}$
- 5: Use Rosen With  $z \text{ min} \leq Z^* \leq z \text{ max}$

`Rn` is number of decimal points.

Rosen algorithm is based on improving an initial value that should honor all constrains at the beginning. Choosing a proper initial value improve the speed of algorithm. Finding an initial value is an issue at kriging with a trend model, because each trend model should be honored by initial value. This algorithm loops until to reach an answer with acceptable accuracy. Algorithm need more time to reach more accuracy. Figure 2 shows the effect of `Rn` parameter in the speed of this algorithm. A domain with 50\*50 grid nodes has been estimated by simple kriging without any constraints. This graph shows a liner correlation between `RN` and the time spent by algorithm.

Figure 3 shows the speed of Rosen algorithm compare to Lagrange method. The same domain with 50\*50 grids nodes was estimated with Rosen and Lagrange methods. Vertical ax is the time that spent by each

algorithm in seconds. The horizontal axis is the number of points that are used to estimate of each block. The  $R_n$  is frozen in 5. This Figure shows the time difference between Lagrange and Rosen methods. By increasing number of points used in estimation, speed of Rosen algorithm is decreased rapidly.

Figure 4, shows the time of rosen algorithm with different constraints. SK has no constraint so it is faster than others . OK use on equal constraint and OK with positive estimate has one equal and one unequal constrains. Using Rosen kriging with positive weights at each estimate point, there is  $N_a$  number of unequal constrains where  $N_a$  is number of data are used in that pint. It is very cleat that Rosen algorithm is very sensitive to number of constrains.

### **Discussion**

Negative weights make Kriging nonconvex that sometimes estimates unacceptable values such as negative grades, etc. There are some methods to solve this problem but Rosen's Gradient method is an exact solution. It can be used with any type of Kriging estimators. This Method can also be used by all nonlinear Kriging estimators such as lognormal Kriging, Gaussian transformation, Indication Kriging and disjunctive Kriging. Because of convexity property of Rosen Kriging there is no need to guess the minimum and maximum data values at Gaussian transformation. With some simple modification it can also be applied to any Kriging estimators as well as Indicator Kriging (IK) and Co-Kriging . This algorithm is significantly faster that alternative non-linear programming algorithms.

### **References**

- Barnes, R. and Johnson, T., 1984, Positive Kriging, in Verly, G., ed., Geostatistics for Natural Resources Characterization: v.1, Reidel, Dordrecht, Holland, pp. 231-244.
- Bazara, M. S. and Shetty, C. M., 1979, Nonlinear Programming: Theory and Algorithms, John Wiley and Sons, pp. 389-399.
- Deutsch, C., 1996, Correcting For Negative Weights in Ordinary Kriging, Computers & Geosciences, 22 (7), 765-733.
- Deutsch, C., and Journel, A., 1992, GSLIB: Geostatistical software library and user's guide: Oxford University Press, New York, 340pp.
- Goovaerts, P., 1997, Geostatistics for Natural Resources Evaluation: Oxford University Press, New York, pp. 174-177
- Herzfeld, C., 1989, A note on programs performing Kriging with nonnegative weights, Mathematical Geology v. 21, no. 3, 391-393.
- Rosen, J. B., 1960, The gradient projection method of nonlinear programming: Part I, linear constraints, SIAM journal, v. 8, 181-217.
- Szidarovszky, F., Baafi, E. Y., and Kim, Y.C., 1987, Kriging Without Negative Weights, Mathematical Geology v. 19, no. 6, 549-559.

```

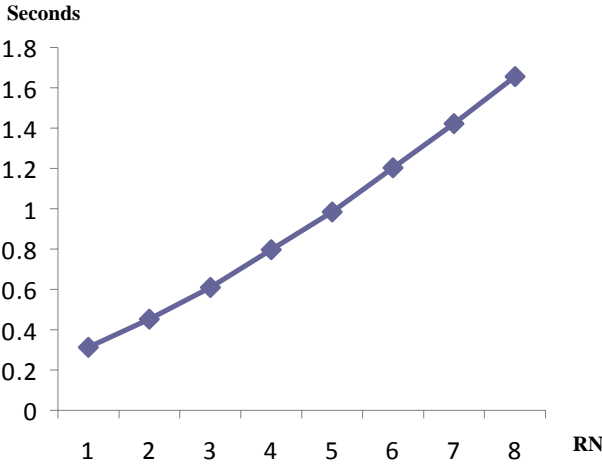
Parameters for KT3D
*****

START OF PARAMETERS:
cluster.dat          -file with data
0 1 2 0 3 0        - columns for DH,X,Y,Z,var,sec var
-1.0e21 1.0e21     - trimming limits
0                  -option: 0=grid, 1=cross, 2=jackknife
xvk.dat            -file with jackknife data
1 2 0 3 0         - columns for X,Y,Z,vr and sec var
0                  -debugging level: 0,1,2,3
kt3d.dbg           -debug file
kt3d.out           -output file
50 0.5 1          -nx,xmn,xsiz
50 0.5 1          -ny,ymn,ysiz
1 0 1             -nz,zmn,zsiz
1 1 1             -x,y and z block discretization
0 20              -min, max data for kriging
0                  -max per octant (0-> not used)
1000.0 1000.0 1000.0 -maximum search radii
0.0 0.0 0.0      -angles for search ellipsoid
1 0               -0=SK,1=OK,2=non-st SK,3=exdrift
0 0 0 0 0 0 0 0  -drift: x,y,z,xx,yy,zz,xy,xz,zy
0                  -0, variable; 1, estimate trend
extdrift.dat      -gridded file with drift/mean
4                  - column number in gridded file
1 0.2             -nst, nugget effect
1 0.8 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
                  -a_hmax, a_hmin, a_vert
1 8 0 60          - stype,RN,zmin,zmax

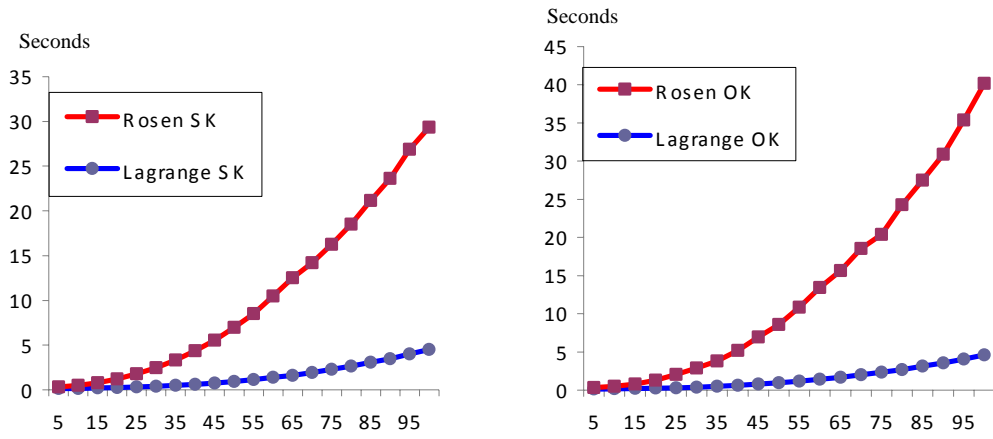
```

**Figure 1.** Rosen parameter file

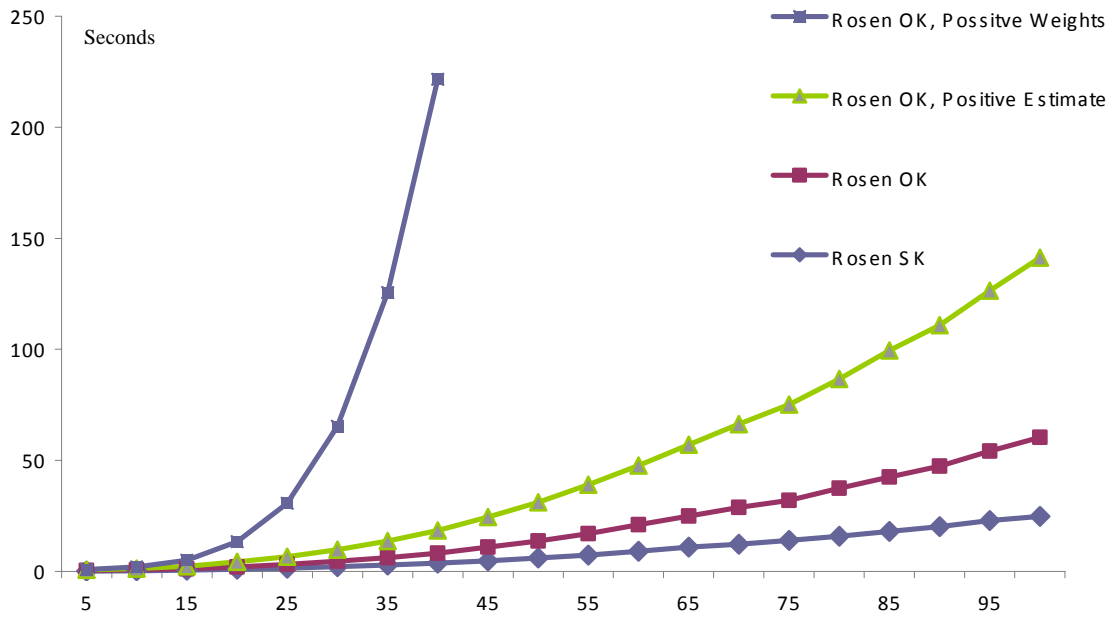
**Rosen, SK W/Different RNs**



**Figure 2.** Impact of Rn paramer in the time of Rosen algorithm



**Figure 3.** Time of Rosen and Lagrange Algorithms; left: Simple Kriging; Right Ordinary Kriging



**Figure 4.** Time of Rosen algorithm with different type of constants