Distance Based Simulation for Generating Additional Realizations

Amir H Hosseini and Clayton V Deutsch

Running dynamic data integration algorithms such as sequential self-calibration requires a large amount of CPU time. It has been observed that not all the resulting realizations support the desired level of measurement error; and therefore a screening is required as a post-processing step. The required screening adds to the time needed for generating the desired number of realizations. An alternative way of expanding the set of Gaussian realizations supporting a particular level of measurement error has been examined in this work. This alternative approach, termed distance-based simulation (Scheidt et al. 2008 and Caers 2008), builds on the concepts of multi-dimensional scaling, KL-expansion, kernel principal component analysis and modeling and simulation in metric and feature spaces. It is observed that this technique needs a very careful post-processing of the realizations to ensure the reproduction of input histogram (especially standard deviation) by the expanded set of realizations. It is also observed that many of the generated realizations may closely resemble the input realizations.

Introduction

Some works in the literature investigate the relationship between the uncertainties in the model parameters and measurement errors and/or predictive error variance. The work of Vecchia and Cooley (1987) is one of the most important early works in this area. The works of Henricks-Franssen and Gomez-Hernandez (2003), McKenna et al. (2003), Moore and Doherty (2005) and Tonkin et al. (2007) are among the recent works in this area for distributed systems. Hendricks-Franssen and Gomez-Hernandez (2003) investigated the impact of measurement error in the stochastic inverse modeling by the SSC method. In their work, they set the tolerance value (J_{tol}) for minimization of the objective function equal to:

$$J_{tol} = n\sigma^2 \quad [1]$$

where, n is the number of head observations, and σ^2 is the variance of head measurement error based on Gaussian noise. Throughout data conditioning, they evaluated the objective function for each updated log-hydraulic conductivity field. The log-hydraulic conductivity field was considered as to be successfully conditioned to head data, if the value of objective function was below the pre-specified tolerance value.

The SSC calibration stops when the value of the objective function is below a pre-specified value, or the improvements in the model fit are insignificant in consecutive iterations, or maximum number of outer iterations exceeds the maximum allowed. These stopping criteria result in conditional realizations that do not show the same level of mismatch to head data. A ranking and screening ensures that the level of mismatch of the conditional realizations with the head observations is consistent with the observations errors. One can define a measure of fit denoted by s as:

$$s^2 = \frac{F}{n} \quad [2]$$

where, F is the value of the objective function while weights are set to the inverse of head error covariance matrix and n is the number of observations. Similar to Equation [1] and the work of McKenna et al. (2003), for every realization subject to conditioning to head data, the value of s should be as close to one as possible to (1) ensure the convergence of the conditioning algorithm, and (2) avoid fitting the noise in observations. This concept can be used in ranking and screening the realizations conditioned to head data by the SSC approach. Instead of running the SSC approach for a large number of realizations and screening, an idea is to select a smaller number of realizations after screening and expand the set of realizations with distance-based simulations.

According to Scheidt et al. (2008) and Caers (2008), using distance-based simulation, one may choose a limited number of realizations that have a desired response and expand the set of realizations with similar responses while being conditioned to static data and reproduce the histogram and variogram. MDS is applied to reduce the dimensionality of the geostatistical realizations from N dimensional space to L dimensional space, where N and L are the number of grid nodes and number of realizations. KL expansion, which relies on eigen-value decomposition of the covariance matrix, can be used to generate new realizations in the metric space that is defined by pair-wise differences in the values of s. When a

non-Euclidean distance (e.g. the measure of fit *s*) is used as the measure of pair-wise difference between the realizations, KL expansion must be implemented in the feature space, where Gaussian/linear type modeling becomes more appropriate. KPCA is used to transform the set of realizations to the feature space. A back-transformation (pre-image problem) is then required to find the corresponding set of realizations in the Cartesian space.

Distance Based Simulation

The distance based simulation starts by generating multiple geostatistical realizations conditioned to all transmissivity and head observations with the values of measure of fit close to one. This can be achieved by conditioning a limited number of realizations to head data using the SSC algorithm, post-processing them, and select L realizations with desired response. Then, the pair-wise distance between the selected realizations is defined as the difference in the values of measure of fit for all realizations. Assuming D = [dij] to be the pair-wise distance matrix defined based on the measure of fit, the centered dot-product matrix B can be calculated by:

$$\mathbf{B} = \mathbf{HAH} \quad [1]$$

where, **H** is the centering matrix and **A** is the equivalent dot-product matrix whose values are calculated by:

$$a_{ij} = -\frac{1}{2} d_{ij}^2 \quad [2]$$

The centering matrix **H** in Equation [A.1] is defined by:

$$\mathbf{H} = \mathbf{I} - \frac{1}{L}\mathbf{1}\mathbf{1}^T \quad \text{with} \quad \mathbf{1} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \end{bmatrix} \quad [3]$$

In the next step, the realizations are mapped from a high dimensional space (*N* dimensions) into a lower dimensional space (up to *L* dimensions). For this purpose, eigenvalue decomposition of the dot product matrix is performed:

$$\mathbf{B} = (\mathbf{HX})(\mathbf{HX})^T = \mathbf{V}\Lambda\mathbf{V}^T \quad [4]$$

where, **X** is an *L* × *N* matrix containing the original *L* realizations as its rows, **V** is the matrix containing the eigenvectors and Λ is the diagonal matrix of eigenvalues. After eigen-value decomposition of the dot-product matrix, the matrix of mapped realizations into a lower dimensional space ($\hat{\mathbf{X}}_d$) is calculated by Equation [5]. The overall process is termed multi-dimensional scaling.

$$\hat{\mathbf{X}}_d = \mathbf{V}_d \Lambda_d^{1/2} \quad [5]$$

where, Λ_d is the matrix containing the *d* largest eigenvalues and $\mathbf{V}_d$ is the matrix of the corresponding eigenvectors. The subscript *d* varies between 1 and the number of realizations *L*. According to Caers (2008), if the Euclidean distance measure is used between realizations with Gaussian distributions, the duality between the dot-product matrix and covariance matrix can be used to generate new realizations as linear combinations of existing realizations through KL expansion:

$$\mathbf{x} = \mathbf{m} + (\mathbf{HX})^T \mathbf{V}_d \frac{1}{\sqrt{L}} \mathbf{y} \quad [6]$$

where, **y**, **m** and **x** represent the vectors of standard Gaussian deviates, the mean of the Gaussian field and the vector of the new Gaussian field which has been calculated as a linear combination of the existing realizations, respectively. When non-Euclidean distance measures are used such as the measure of fit, it is observed that the problem becomes non-Gaussian and the new realizations generated by KL expansion do not reproduce the desired response being the measure of fit close to one. To overcome this problem, Scheidt et al. (2008) proposed (1) transforming the realizations into feature space using a radial basis function (RBF), (2) performing KL expansion in the feature space and (3) back transforming the realizations to Cartesian space by solving the pre-image problem.

Transforming the realizations to the feature space ($\hat{\mathbf{x}}_{d,i} \mapsto \boldsymbol{\varphi}(\hat{\mathbf{x}}_{d,i})$) is a complex problem, and according to Scheidt et al. (2008), involves a high dimensional multivariate function $\boldsymbol{\varphi}$ whose determination is not easy. However, as shown in Equations [4], [5] and [6], to perform KL expansion, we

only need eigenvalue decomposition of the matrix of dot-product in the feature space, which can be calculated by kernel RBF as:

$$K_{ij} = k\left(\hat{\mathbf{x}}_{d,i}, \hat{\mathbf{x}}_{d,j}\right) = \left\langle \boldsymbol{\varphi}\left(\hat{\mathbf{x}}_{d,i}\right), \boldsymbol{\varphi}\left(\hat{\mathbf{x}}_{d,j}\right) \right\rangle = \exp\left(-\frac{\left(\hat{\mathbf{x}}_{d,i} - \hat{\mathbf{x}}_{d,j}\right)^T \left(\hat{\mathbf{x}}_{d,i}, \hat{\mathbf{x}}_{d,j}\right)}{\sigma^2} \right) \quad [7]$$

where, ? is the tuning parameter. The K_{ij}'s are collectively form the components of a matrix termed the Gram matrix **K**. In the feature space, new realizations can be generated based on the existing transformed realizations through KL expansion:

$$\mathbf{C}_\phi = \frac{1}{L}\boldsymbol{\Phi}\boldsymbol{\Phi}^T = \mathbf{V}_{C,\phi}\boldsymbol{\Lambda}_{C,\phi}\mathbf{V}_{C,\phi}^T \;\Rightarrow\; \boldsymbol{\varphi}_{new}(\mathbf{x}) = \mathbf{V}_{C,\phi}\boldsymbol{\Lambda}_{C,\phi}^{1/2}\mathbf{y} \quad [8]$$

where, $\boldsymbol{\Phi}$ is the matrix of unknown existing transformed realizations into the feature space, $\mathbf{C}_\phi$ represents the covariance matrix of the transformed realizations, $\mathbf{V}_{C,\phi}$ is the matrix that involves the eigenvectors for covariance matrix, and $\boldsymbol{\Lambda}_{C,\phi}$ is the corresponding diagonal matrix of the eigenvalues. Similar to Equations [4], [5] and [6], the duality between the dot product matrix and the covariance matrix in the feature space can be used to derive the relationship between their eigenvalues and eigenvectors:

$$\boldsymbol{\Lambda}_{C,\phi} = \frac{1}{L}\boldsymbol{\Lambda}_\phi \quad , \quad \mathbf{V}_{C,\phi}^T = \boldsymbol{\Phi}^T\mathbf{V}_\phi\boldsymbol{\Lambda}_\phi^{1/2} \quad [9]$$

where, $\mathbf{V}_\phi$ is the matrix involving the eigenvectors for the dot-product matrix, and $\boldsymbol{\Lambda}_\phi$ is the corresponding matrix of eigenvalues. Combining Equations [8] and [9], new realizations in the feature space can be expressed as linear combination of existing realizations also in the feature space:

$$\boldsymbol{\varphi}^{new}(\mathbf{x}) = \boldsymbol{\Phi}^T\boldsymbol{\alpha} \quad [10]$$

where, a represents the vector of coefficients whose components can be calculated by:

$$\alpha_i = y_i\lambda_{\phi,i}^{1/2}\sum_{j=1}^{d} v_{\phi,i,j} \quad \text{with} \quad 2 \le d \le L \quad \text{and} \quad i = 1, \ldots, L$$

where, $v_{\phi,i,j}$ represents a component of the matrix of the eigenvectors for the dot-product matrix, y_i is a component of the vector of Gaussian deviates, and $\lambda_{\phi,i}$ is a component of diagonal matrix of eigenvalues for the dot-product matrix. The new set of realizations generated in the feature space must be back-transformed to Cartesian space. As the inverse of $\boldsymbol{\varphi}$ is not explicitly known, this back-transformation translates into an ill-posed inverse problem (also known as pre-image problem) that is formulated as an optimization problem:

$$\hat{\mathbf{x}}_{d,new} = \underset{\hat{\mathbf{x}}_{d,new}}{\arg\min} \left\| \boldsymbol{\varphi}\left(\hat{\mathbf{x}}_{d,new}\right) - \boldsymbol{\Phi}^T\boldsymbol{\alpha} \right\| \quad [11]$$

Following the recommendations by Schoelkopf and Smola (2002), the fixed-point method can be used to solve the pre-image problem. Implementation of the fixed point method results in the expression for back-transformation of the newly generated realizations from the feature space to the MDS space (up to *L* dimensions):

$$\hat{\mathbf{x}}_{d,new} = \sum_{i=1}^{L} \beta_i\left(\hat{\mathbf{x}}_{d,i}\right)\hat{\mathbf{x}}_{d,i} \quad \text{with} \quad \beta_i\left(\hat{\mathbf{x}}_{d,i}\right) = \frac{\alpha_i k'\left(\hat{\mathbf{x}}_{d,new}, \hat{\mathbf{x}}_{d,i}\right)}{\sum_{j=1}^{L} \alpha_j k'\left(\hat{\mathbf{x}}_{d,new}, \hat{\mathbf{x}}_{d,j}\right)} \quad [12]$$

where, k' is the derivative of the kernel RBF in Equation [A.7]. According to Scheidt et al. (2008), one can perform unconstrained optimization to find the newly generated realizations into the *N* dimensional Cartesian space. The unconstrained optimization is appropriate for Gaussian fields and is performed through using the same weights of Equation [A.12] to the realizations in the Cartesian space:

$$\mathbf{x}_{new} = \sum_{i=1}^{L} \beta_i\left(\hat{\mathbf{x}}_{d,i}\right)\mathbf{x}_i \quad \text{with} \quad \beta_i\left(\hat{\mathbf{x}}_{d,i}\right) = \frac{\alpha_i k'\left(\hat{\mathbf{x}}_{d,new}, \hat{\mathbf{x}}_{d,i}\right)}{\sum_{j=1}^{L} \alpha_j k'\left(\hat{\mathbf{x}}_{d,new}, \hat{\mathbf{x}}_{d,j}\right)} \quad [13]$$

In Equations [12] and [13], the weights sum to one, which ensures the reproduction of conditioning static data at the data locations. Also, it is observed that the calculation of the weights depends on the new realizations (in the MDS space) themselves. Thus, an iterative procedure is employed in this work to quantify the weights. To find the weights to generate a new realization, one can start from one of the existing realizations (mapped into the MDS space ($\hat{\mathbf{x}}_{d,j}$)) each time as an initial guess for $\hat{\mathbf{x}}_{d,new}$, and calculate a new vector of weights . The new vector of weights is then used to find an updated $\hat{\mathbf{x}}_{d,new}$ using Equation [A.12]. This process is repeated until the difference in $\hat{\mathbf{x}}_{d,new}$ in consecutive iterations becomes small and the convergence is achieved. Then, the same weights are used in Equation [A.13] to find the desired number of output realizations, using a given number of input realizations in the Cartesian space. The methodology does not always result in governance, when calculating the weights. Also, some of the generated realizations may closely resemble the input realizations. So, a post-processing is almost always required. Despite these shortcomings, given the fact that the algorithm is significantly faster than conditioning new realizations to head data by inverse modeling, it can be considered a useful methodology.

Example

Figure 1 shows a 2D reference log hydraulic conductivity field, the associated steady-state piezometric head response, and the locations of sampling points. The simulation domain is 250 m by 160 m and the squared shape grid blocks are 2.0 m by 2.0 m. The reference log hydraulic conductivity field has a mean of -10.1, standard deviation of 1.2 both in natural logarithm units ($\log_e$ m/s) and a spatial correlation defined by a spherical variogram with a nugget effect equal to 0.1 and a range of 36.0 m. The boundary conditions for the flow field include fixed head boundary conditions at the north and south of the site with constant head values of 4.5 m and 2.0 m, respectively; and no-flow boundary conditions at the west and east boundaries. There are 100 head observation locations (black circles) and 18 permeability measurement locations (white circles). Three data sets for piezometric heads have been created by introducing two different levels of Gaussian noise (= 0.10m, 0.15m) to the piezometric head values sampled from the reference head field shown in Figure 1-b. Two sets of 1000 hydraulic conductivity realizations conditioned to hydraulic conductivity data only are generated by sequential Gaussian simulation program of GSLIB (Deutsch and Journel 1998). Figure 2 shows the histograms of the measure of fit *s* for the two sets of hydraulic conductivity realizations calculated based on different levels of error in the head observations, before conditioning to head data by the SSC. Figure 3 shows the histograms of the values of *s* for the two ensembles of realizations after conditioning to head data by the SSC. In the implementation of the SSC, a total of 126 master points are used (roughly two master points per correlation range in each direction), a maximum of 25 outer iterations are allowed, and the damping parameter and the minimum relative tolerance (normalized to the initial value of the objective function) are set to 0.2 and 0.01, respectively. Also, the minimum difference of objective function in two consecutive iterations, and the maximum number of times that the difference of objective function is smaller than the pre-specified value are set to 0.005 and 10, respectively. The minimum relative tolerance has been intentionally set to a small number (0.01) to ensure that the optimization process searches for the best possible fit to the observed heads to investigate the effects of different levels of noise on the histograms of *s*. In Figures 2 and 3, it can be observed that: (1) when the standard deviation of measurement error is large (noisy data set), there may be some realizations that have their measure of fit close to one (Figure 5-3-c) and therefore acceptable even before conditioning to head data; (2) when the standard deviation of measurement error is small, there may only be a small number of conditioned realizations (by the SSC) that satisfy the requirement of *s* close to one (Figure 3-a); and (3) setting the target value of the objective function to a very small value when the observed head data is too noisy may result in conditional realizations that have a good fit (over-fitted) to noisy data set but their value of *s* deviates from one.

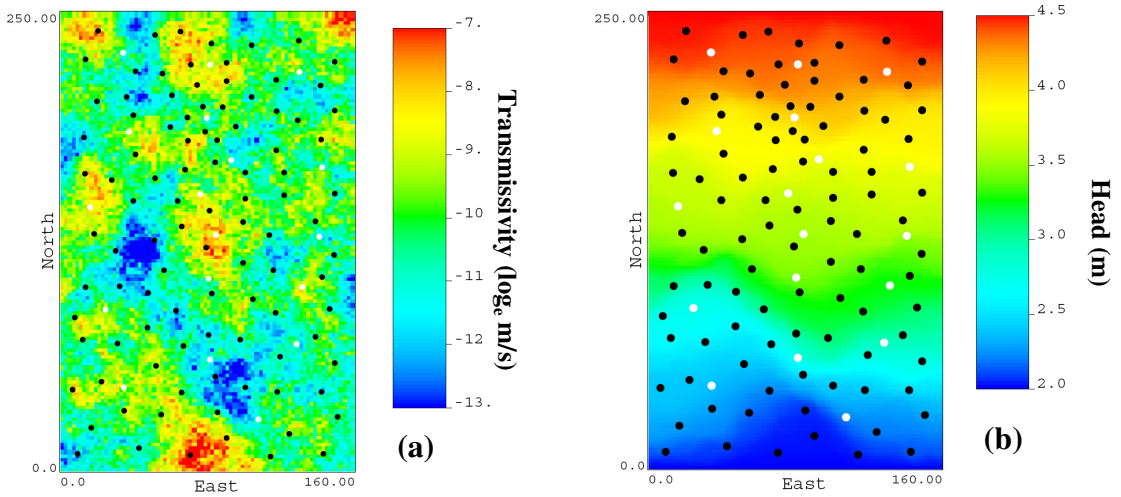

Figure 1: (a) Reference hydraulic conductivity field, and (b) the associated piezometric head response. There are 100 head observation locations (black circles) and 18 hydraulic conductivity measurement locations (white circles).

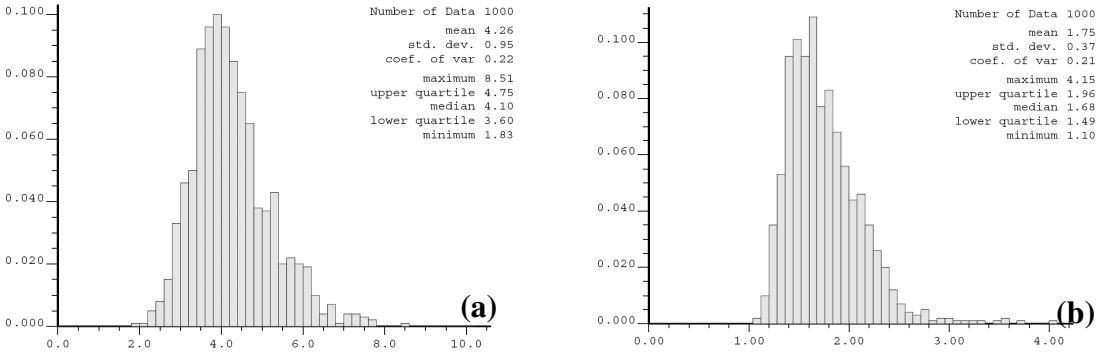

Figure 2: Histogram of the measure of fit *s* for the ensembles of realizations before conditioning to head data for measurement error standard deviations of (a) □ = 0.10 m and (b) □ = 0.15.

Running the SSC program requires a large amount of CPU time. In the above example, the required CPU time for running the SSC algorithm for 1000 realizations under steady-state conditions (with known boundary conditions) was roughly equal to 62 hours on a Dell Precision PWS470 workstation. Under transient flow conditions, the required CPU time can be significantly larger. Therefore, generating a large number of realizations, conditioning them to head data by the SSC, and accept/reject them based on the closeness *s* to one can be quite time-consuming. Using distance-based simulation, one may choose a limited number of realizations that have a desired response and may expand the set of realizations that have similar responses while being conditioned to static data and reproduce (some of) the input statistics.

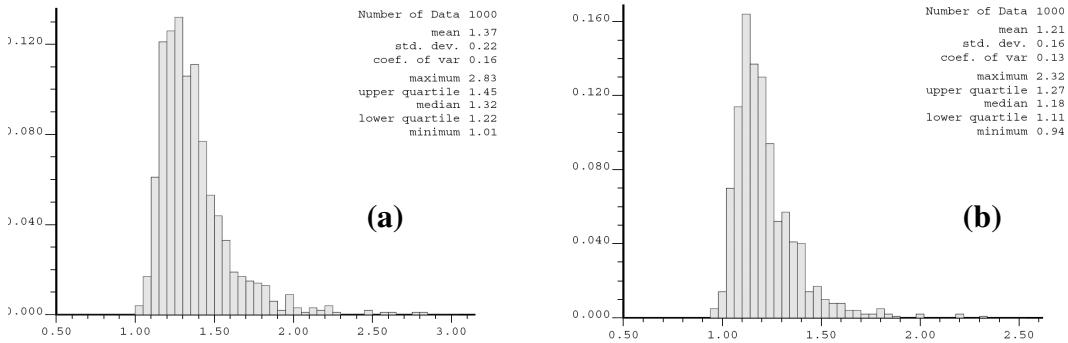

Figure 3: Histogram of the measure of fit *s* for the ensembles of realizations after conditioning to head data for measurement error standard deviations of (a) σ = 0.10 m and (b) σ = 0.15.

Figures 4-a, b, c show the projection of the 1000 Gaussian realizations into a 3D MDS space based on a Euclidean distance measure, measure of fit as the distance measure, and after transforming the non-Euclidean projection of the realizations into the feature space, respectively. It can be observed that the disorganized cloud of points in Figure 4-1-b becomes close to linear, which is more appropriate for modeling (KL expansion), after transforming to the feature space.

When the level of measurement error is small, there may be only a small number of realizations that are accepted. Given the low acceptance rate of the realizations, a faster alternative would be the distance-based simulation (explained above) to expand the set of acceptable realizations. As an example, two sets of 50 realizations that their values of measure of fit are closest to one are selected among the realizations that are conditioned to head data (by SSC) with error standard deviations of 0.10 m and 0.15 m. It should be noted that all the realizations conditioned to head data by SSC have Gaussian distributions. The sets of realizations are used as input for the distance based simulations, and the sets of acceptable realizations are expanded to 500 realizations. The difference in the values of measure of fit is applied as the non-Euclidean measure of distance. Figure 5 shows the histograms of the values of the measure of fit for the expanded sets of realizations. Figure 6 shows two example transmissivity realizations generated by distance based simulations, Figure 7 shows the variogram reproduction for 20 realizations generated by the SSC algorithm and distance based simulation in North-South direction. Comparing the realizations in Figure 6 to Figure 1-a, the SSC algorithm and subsequent distance based simulation reproduce the overall structure of the reference transmissivity field.

Codes

Two codes have been developed: **mds** and **klesim**. The code **mds** takes multiple realizations and maps them in a lower dimensional space based on a given measure of distance between the realizations. The dimensionality of the target space depends on the number of input realizations. Figure 8 shows the parameter file for the code. In Figure 8, line 1 is the name of the file with input realizations; line 2 is the column number for the attribute; line 3 is the number of realizations, line 4 is x, y, z discretization; line 5 is the flag for Euclidean/non-Euclidean distance; line 6 is the name of the file with distance measure; line 7 is the column number for the distance measure; line 8 is the target dimension for mapping (with a maximum of *L*); and line 9 is the output file name with the mapped realizations.

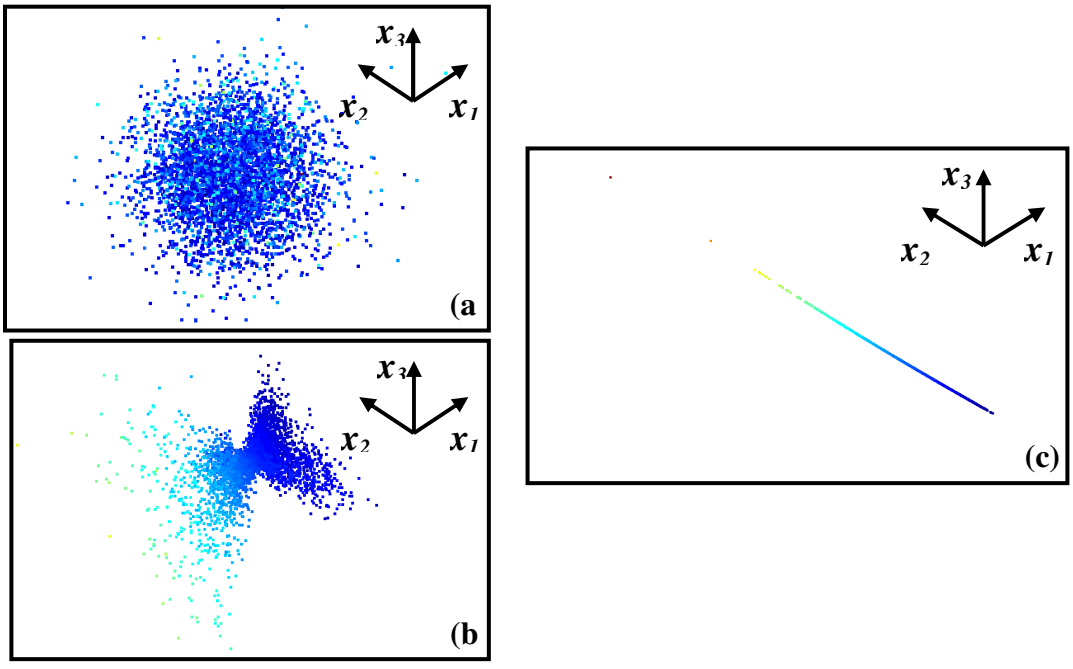

Figure 4: The 3D mapped realizations into the MDS space (a) based on Euclidean distance between the realizations, (b) based on non-Euclidean distance (difference in measure of fit), and (c) after transforming to the feature space by kernel RBF.

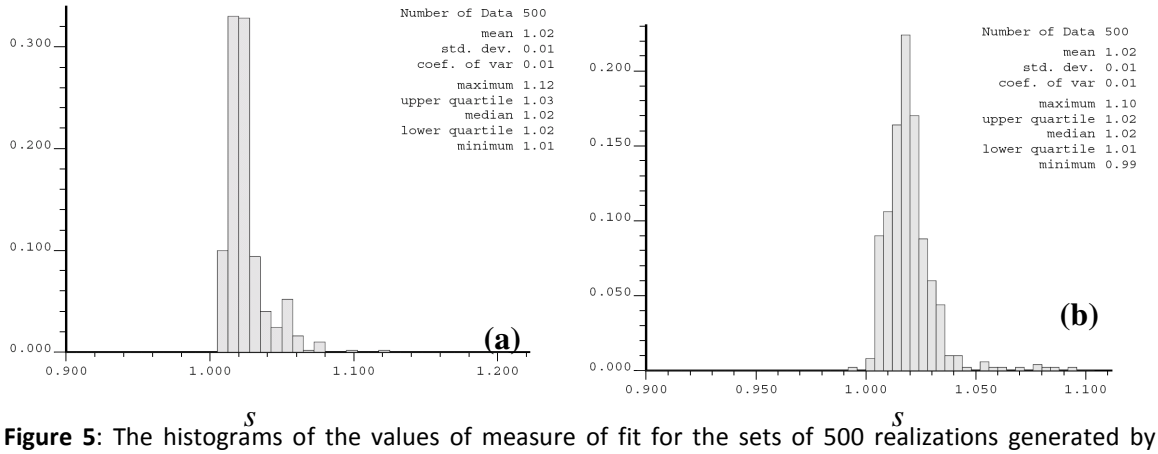

Figure 5: The histograms of the values of measure of fit for the sets of 500 realizations generated by distance based simulation for (a) standard deviation of error σ equal to 0.10m, and (b) standard deviation of error σ equal to 0.15m.

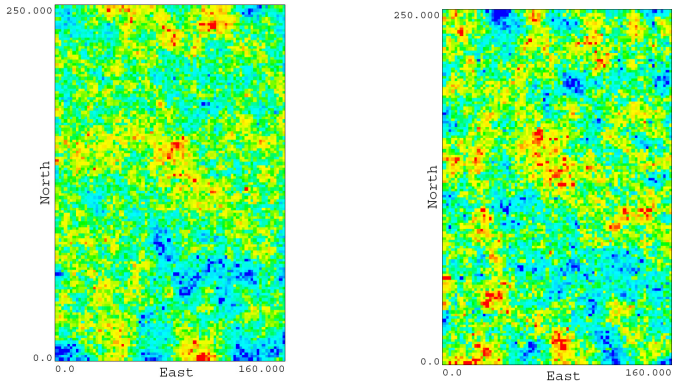

Figure 6: Two example realizations generated by distance-based simulations for a standard deviation of error σ equal to 0.15m.

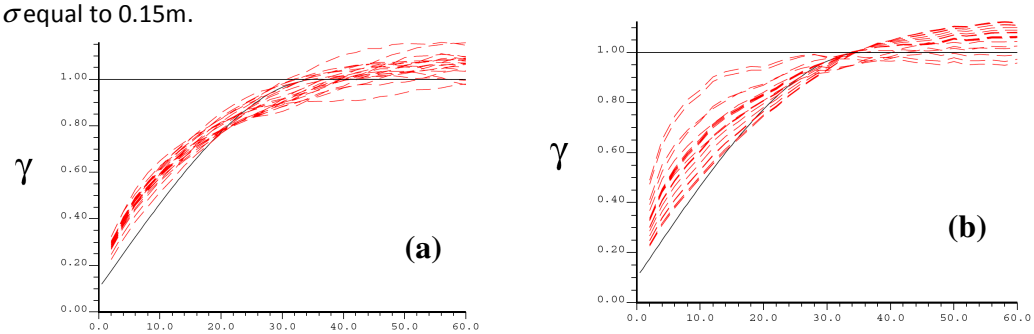

Figure 7: The variogram reproduction check for (a) the first 20 realizations generated by the SSC algorithm and (b) the first 20 realizations generated by distance-based simulation.

Figure 9 shows the parameter file for the code **klesim**. In Figure 9, line 1 involves the name of the file with input realizations; line 2 involves the column number for the attribute; line 3 involves the number of realizations, line 4 involves x, y, z discretization; line 5 involves the flag for Gaussian/non-Gaussian realizations; line 6 involves the flag for Euclidean/non-Euclidean distance; line 7 involves the name of the file with distance measure; line 8 involves the column number for the distance measure; line 9 involves the flag for reporting the feature space; line 10 involves the filename with realizations in the feature space; line 11 involves the number of output realizations; line 12 involves the file name for output realizations; line 13 involves the random number seed; line 14 involves the flag for controlled

dimensioning; line 15 involves the target dimension (if designated to be less than *L*); line 16 involves the value of the tuning parameter for Radial-basis function; and line 17 involves the maximum number of iterations and the tolerance value for the fixed point method.

```
                  Parameters for MDS
                  ******************

   START OF PARAMETERS:
1  selected_realiz_NEU.out         -file with realizations
2  1                               -   column for attribute
3  50                              -   number of realizations
4  80   125   1                    -   nx,ny,nz
5  1                               -0=Euclidean,1=non-Euclidean
6  flow2d_mr-SELECT.out            -   file with distance measure(ieu=1)
7  3                               -   column for distance measure
8  3                               -target dimension for mapping
9  mds.out                         -file with mapped realizations
```

Figure 8: The parameter file for the code **mds**.

```
                  Parameters for KLESIM
                  *********************

    START OF PARAMETERS:
1   selected_realiz_NEU.out       -file with input realizations
2   1                             -   column for attribute
3   50                            -   number of input realizations
4   80   125   1                  -   nx,ny,nz
5   0                             -   Gaussian(igu=0),non-Gaussian(igu=1)
6   1                             -Euclidean(ieu=0),non-Euclidean(ieu=1)
7   flow2d_mr-SELECT.out          -   file with distance measure(if ieu=0)
8   3                             -   column for distance measure
9   0                             -report the feature space(if igu=1)
10  kernel_RBF.out                -   file with realizations in feature space
11  20                            -number of output realizations
12  klesim.out                    -file with output realizations
13  69079                         -random number seed
14  1                             -control dimensioning(0=no,1=yes)
15  2                             -   target dimension for MDS(if icd=1)
16  1.0                           -RBF tuning parameter
17  30    0.01                    -fixed point iterations:number,tol.
```

Figure 9: The parameter file for the code **klesim**.

Conclusions

Although distance based simulation may be considered useful in expanding the set of acceptable realizations, there are some serious issues that must be taken into account. First, solving the pre-image problem by fixed point method is only acceptable when the realizations are Gaussian; and even Gaussian realizations do not ensure the convergence of the approach. Also, in terms of the reproduction of basic statistics, the generated realizations show a reasonable reproduction of variogram in most cases. The histogram reproduction, however, is not guaranteed. Although the Gaussian shape of the histogram and its mean are almost always reproduced, the standard deviation of output realizations may considerably over- or under-estimate the true standard deviation. It is also observed that many of the generated realizations may closely resemble a few of the input realizations. Due to these issues, and based on Author's experience, the distance-based simulation (in the current level of development) should be used with care and post-processing (acceptance/rejection) of the generated realizations is needed.

References

Caers, J. (2008), Distance-Based Random Field Models: Theory and Applications, *8th International Geostatistics Congress*, Santiago, Chile, December 1-5

Deutsch, C.V., and A.G. Journel (1998), *GSLIB: Geostatistical Software Library and User's Guide*, Oxford University Press, New York, NY

Scheidt C., K. Park, J. Caers (2008) Defining a random function from a given set of model realizations, *8th International Geostatistics Congress*, Santiago, Chile, December 1-5

Schoelkopf, B. and A. Smola (2002), *Learning with Kernels*, MIT Press, Cambridge, MA