

Calculation of Feasible Domain of Translation for Conditional Finite Domain Uncertainty Calculation

Behrang Koushavand and Clayton V. Deutsch

One of the main goals of geostatistics is to quantify uncertainty. Histograms are one of the most important parameters in geostatistical modeling. The mean of the histogram is perhaps the most important parameter because it has a direct effect on calculated resources and reserves. The conditional finite domain (CFD) is a new technique to get uncertainty in the mean that accounts for the size of the domain and the local conditioning data. The CFD technique requires translating the data configuration to assess uncertainty, but the calculation of the feasible domain of translation is a non-trivial problem. An penalty function based approach is developed in this paper. The original data are translated in this domain to generate number of configurations. For each configuration, the original data are used to generate conditional realizations. Histogram uncertainty is calculated based on the translated realizations. Two case studies are presented to demonstrate the uncertainty in the histogram with a focus on the uncertainty in the mean. The results of the bootstrap, spatial bootstrap and conditional finite domain are compared. The results show that conditioning data changes the uncertainty on parameters such as mean.

1. Introduction

The statistical parameters computed from a relatively small data set are inevitably associated with uncertainty. This uncertainty has a significant effect on large scale uncertainty computed from geostatistical realizations. The Conditional Finite Domain (CFD) approach (Babak and Deutsch, 2009) is used to generate conditional realization inside the domain. The sample data configuration is fixed and translated to generate new data configurations inside the domain. New data are simulated conditional to the original data values. Parameters such as the mean, variance and variogram are calculated using the simulated data (Koushavand and Deutsch, 2009). This procedure is repeated to assess uncertainty in the desired input parameters.

A significant challenge with the CFD approach is that if the vector of transition is too small, the uncertainty appears to be unrealistically low; and if the translation vector is too large, the uncertainty is independent from data values and tends to the result of the spatial bootstrap.

It is important to find a feasible domain of translation for the successful application of CFD. In this study, a simple penalty function is defined and calculated for all sample points based on a set of translation vectors. The expected value of the objective function is calculated and a threshold is applied to define the feasible domain of translation.

2. Methodology

A penalty function was defined and the objective function is the expected value of this penalty function over all points. The goal is to find a domain of translation that minimizes the objective function. The penalty function was kept simple so as to avoid any extra parameters. This is important for practical purpose that the domain of optimal translation is found automatically in a robust manner.

To calculate the penalty function, two points for each translation are required: the closest location to the translated location and the closest location to the point being translated. These points are found inside a reasonable search sector along the direction of translation. Figure 1 shows the location being translated and the search sector. Three vectors are defined as bellow:

- h_1 : Proposed translation vector
- h_2 : Vector from $u+h_1$ to the closest data (inside the search sector)
- h_3 : Vector from u to the closest data in the direction of h_1 (inside the search sector)

The penalty function has two components. The first component decreases the objective function when the vector of translation is large, this allows for maximum translation of u . The penalty decreases by increasing the distance of the translated location from the closest point to the location being translated. The definition of first component of penalty function is defined:

$$P_1(h_1, h_2) = \begin{cases} 0 & \text{if } h_2 \text{ undefined} \\ -|h_1|.a_1 & |h_1| \leq |h_2| \\ -|h_2|.a_1 & |h_1| > |h_2| \end{cases} \quad (1)$$

Figure 2 shows the first component of the objective function. Translation vector h_1 begins at 0, because h_1 is less than h_2 , the first part of P_1 (Figure 2a) decreases the penalty function. When $h_1 > h_2$, the second part of P_1 (Figure 2a) increase the penalty. The slope of both lines is defined by a_1 .

The second component limits the vector of translation, because we don't want to translate far beyond the domain of our data. This component doesn't add any penalty to the function for a certain distance then increases the objective function. Therefore the penalty function for second component is defined as:

$$P_2(h_1, h_3) = \begin{cases} (|h_1| - |h_3|).a_2 & |h_1| > |h_3| \\ 0 & \text{otherwise (including if } h_3 \text{ undefined)} \end{cases} \quad (2)$$

Figure 3 shows the second component of penalty function. There are two parameters called a_1 and a_2 related to first and second component of penalty function. These two parameters control the slop of each of component. To obtain resalable results, a_1 should be selected to be less than a_2 . The steps of this algorithm are:

- 1- Start with a vector h_1
- 2- Set Obj=0
- 3- For each location u_i , put the tail of vector h_1 on this location.
- 4- Search for closest location to u_i and closest location to the head of the vector.
- 5- Calculate P_1 and P_2 .
- 6- Obj=Obj+(P_1+P_2).
- 7- Go to step 3 until all locations has been visited.
- 8- Write the value of obj for that specific vector on the output file.
- 9- Choose another vector with different length and direction, go to step 2.

A GSLIB (Deutsch and Journal, 1998) like program called CDF_OPT has been designed to calculate the final objective value which is the expected value of the objective function.

3. Small case study

Figure 4 shows 9 synthetic points, each separated by 10 units. To test the methodology, only point 1 is translated east-west. a_1 is set to 1 and a_2 is 1.1. Figure 5 shows the result of this translation, the dashed blue line is the first component of penalty function. The dashed red line is the second component and the solid green line is the objective function which is a summation of the first and second components. Because there is no location to the left of point 1 with an EW direction, the objective function is 0. To the right of point 1, the first component increases. From point 1 to middle distance of point 1 and 2 which is 5 units; the h_1 is still effective and decreases the first component. From this point h_2 is effective and increase the objective function. This scenario is happened at distance between points 2 and 3. However, after point 3, first component decreases with the same slope. When the translated point is close to another sample, the first component adds some penalty values to the objective function. For the second component (dashed red line), there is no penalty between points 1 and 2, but for the rest, second component add some penalty values and enforce the objective function to be: too far is undesirable. It is reasonable that for the translation at East-West direction, the optimal translation should be somewhere between location 1 and 2.

Figure 6 shows the objective function at different directions.

Table 1 shows the grid definition for head of translation vectors. The tails of vectors were set on the each number and heads were set to center of grid locations described as

Table 1. Figure 6 shows the expected value of objective function for each translation. Figure 7 illustrates the cross section at EW direction. The line crosses the origin. As is expected, distance of 5 units is the optimal translation.

4. Case Study 1

The thickness variable of “Red.dat” data set has been used. Figure 8 shows the location map and histogram of this data. The points have been translated to be centered at origin.

Figure 9 shows the parameter file of *CFD_opt*, *a1* and *a2* are set to 1 and 1.1 respectively. Figure 10 illustrates the map of objective function. At top, all possible translations that the objective function is negative has been presented. Because we are interested in negative values, all positive values of objective function have been trimmed. At the bottom of the same graph we eliminate all translations that the objective function is larger than -10. Histogram of values of the objective function are shown in Figure 11. Mean is -10. For this specific case study cut-off equal to the mean (-10.00) is chosen as the maximum value of objective function, which is close to the p50 or the median. The cross section of objective function through the EW direction is presented at Figure 12.

The program called *SGS_CFD* which is modified version of *SGSIM* (Deutsch and Journel, 1998) has been created to generate conditional realization at a feasible domain of translation at a finite domain. This program reads data file and feasible translations vectors that are generated by *CFD_OPT*. Figure 13 shows the parameter file for *SGS_CFD*. An isotropic spherical variogram model with 250 unit range and 15% nugget effect was used. Based on parameter file 100 random translations have been chosen by this program. Figure 10, bottom graph shows the tail locations of translation vectors that *SGS_CFD* chooses for translation of data.

To compare CFD approach with Spatial Bootstrap (SB), both methods were applied.

Figure 14 shows the cumulative distributions of 1000 realizations and reference data for both SB and CFD methods. The standard deviation of mean for SB and CFD respectively are 1.12 and 0.433. The CFD method has less uncertainty on mean than SB for this specific case study.

Figure 15 shows the histogram of mean of realizations that are generated by these two methods. It is clear that uncertainty of mean for CFD approach is less than SB.

4. Case Study 2

‘Cluster.dat’ data set was used as the second case study. Locations of samples and the histogram of variable of interest are presented in Figure 16. The parameter file for *CFD_OPT* is shown in Figure 17. Figure 18 shows the map of the objective function. Figure 19 shows the CDF of the objective function. All positive values are trimmed. The mean value is -0.85. The mean value, -0.85, has been chosen as a cut-off for the objective function. Figure 18 lower shows the map of all values below this cut-off. Figure 20 shows the *SGS_CFD* parameter file. An isotropic variogram with range of 10 units and a 20% of nugget effect has been used. Figure 18 bottom shows the random vectors inside the feasible domain of translation that are chosen by *SGS_CFD*. 100 translation vectors were randomly chosen from this domain and 100 realizations were generated for each translation. Figure 21 shows the histogram uncertainty with the spatial bootstrap (above) and CFD (below). The histogram of means for 10,000 realizations is presented in Figure 22. For this case study the uncertainty on mean with CFD approach is less than SB technique.

5. Conclusions

Variance of mean (Kitanidis, 1997) without any spatial dependency is calculated by eq. (3):

$$\sigma_{mean} = \frac{\sigma_{data}}{\sqrt{n}} \quad (3)$$

The uncertainty on the mean when data are spatially dependent was calculated by Spatial Bootstrap. CFD is a new technique to calculate uncertainty for input parameters, such as the mean, based on location of data, spatial correlation between data and original data values. Original data values are used to simulate a set of locations with the same data configurations but randomly translated.

Table 2 shows the comparison of three different approaches to calculate uncertainty in the mean. The spatial bootstrap technique always shows high uncertainty on mean. CFD method has less uncertainty than the SB for both data sets. For the Red data set, CFD has higher uncertainty on the mean than the bootstrap, but the interesting result is CFD has less uncertainty for cluster data than both bootstrap and SB methods.

In this paper, a methodology has been presented to find the optimal domain of translation. This domain is very important when implementing the CFD approach. For large vectors of translation, the CFD approach tends

towards the spatial bootstrap approach, and for small vectors the CFD approach shows unrealistically small parameter uncertainty. Therefore a domain of optimal translation is critical to a sound CFD implementation.

In the proposed methodology a penalty function is used to determine the appropriate domain of translation. There are two components for this function. Expected values of the penalty function were calculated and for each vector of translation a threshold was applied to find the optimal domain.

References

Babak, O., Deutsch, C., 2009, *Accounting for Parameter Uncertainty in Reservoir Uncertainty Assessment: The Conditional Finite-Domain Approach*. Natural Resources Research 18, 7-17.
 Deutsch, C.V., Journel, A.G., 1998, *GSLIB : geostatistical software library and user's guide*, 2nd Edition. Oxford University Press, New York, 369 p.
 Kitanidis, P.K., 1997, *Introduction to Geostatistics*. Cambridge University Press.
 Koushavand, B., Deutsch, C.V. 2009. *Conditional Finite Domain Based Variogram Uncertainty*. In 2009 CCG Annual Report Papers, Deutsch, C.V., ed. (Edmonton, Centre for Computational Geostatistics), pp. 411-411 to 411-416.

Table 1. Grid definition for head locations of translation vectors

	number	min	size
x	101	-50	1
y	101	-50	1

Table 2. Mean uncertainty of two case studies with three deferent approaches: Bootstrap, SB and CFD

	RED		Clustered	
	#	%	#	%
σ_{data}	3.97	100	6.7	100
σ_{data} / \sqrt{n}	0.486	12.24	0.513	7.66
σ_{SB}	1.121	28.237	0.960	14.33
σ_{CFD}	0.433	10.907	0.652	9.73

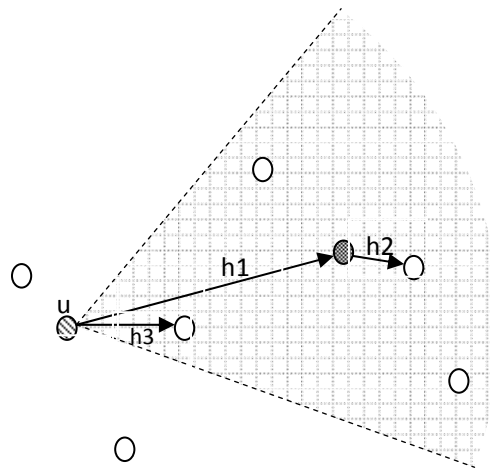


Figure 1. Translation vectors and search sector

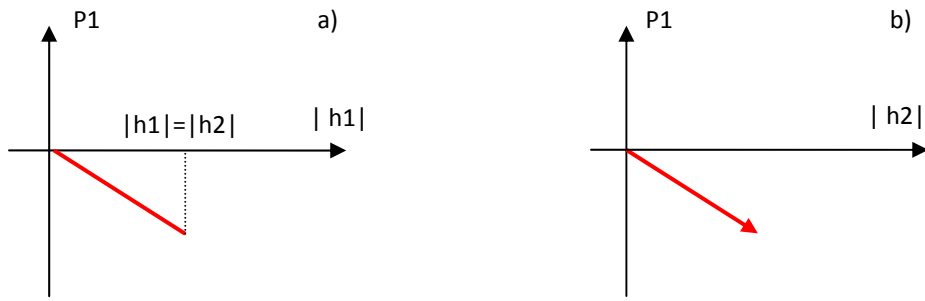


Figure 2. First component of penalty function. A: when $|h_1| \leq |h_2|$ and b: $|h_1| > |h_2|$

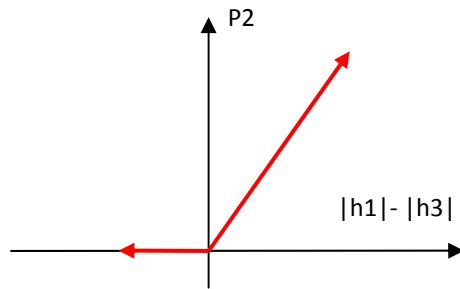


Figure 3. Second component of penalty function

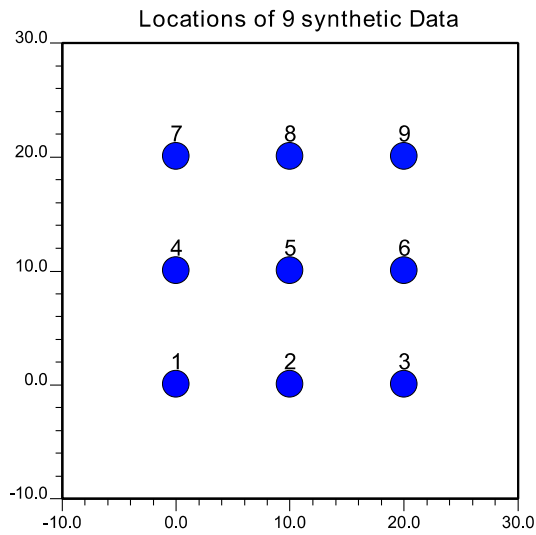


Figure 4. Location map of 9 synthetic data

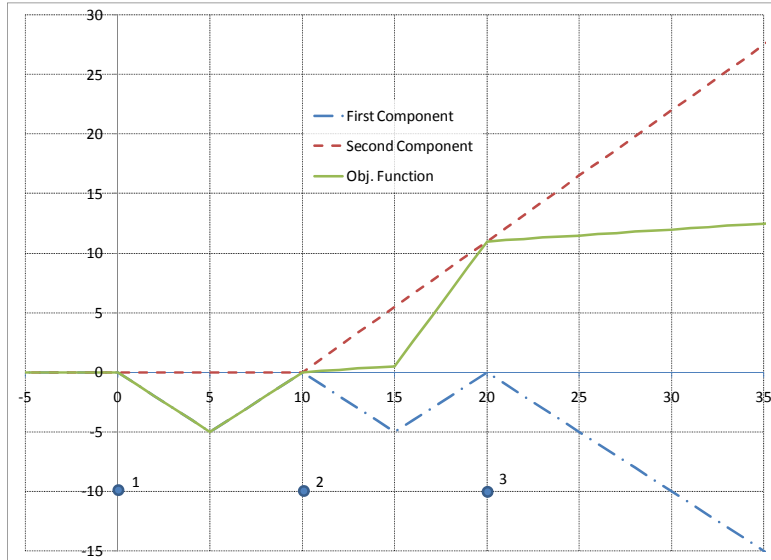


Figure 5. First Component (dashed blue line), Second Component (dashed red line) and objective function (solid green line) are calculated for point number 1 relative to Figure 4

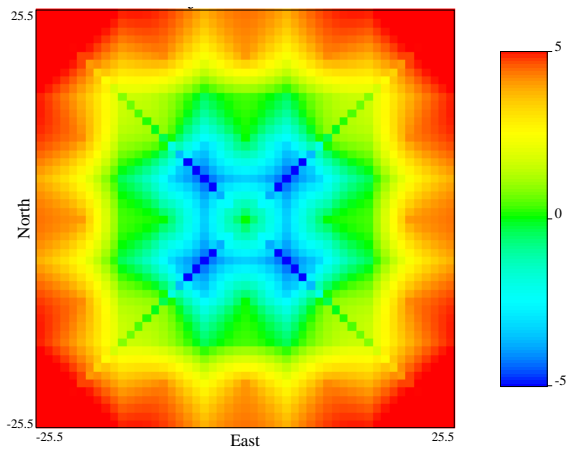


Figure 6. map of objective function for the 9 point case study

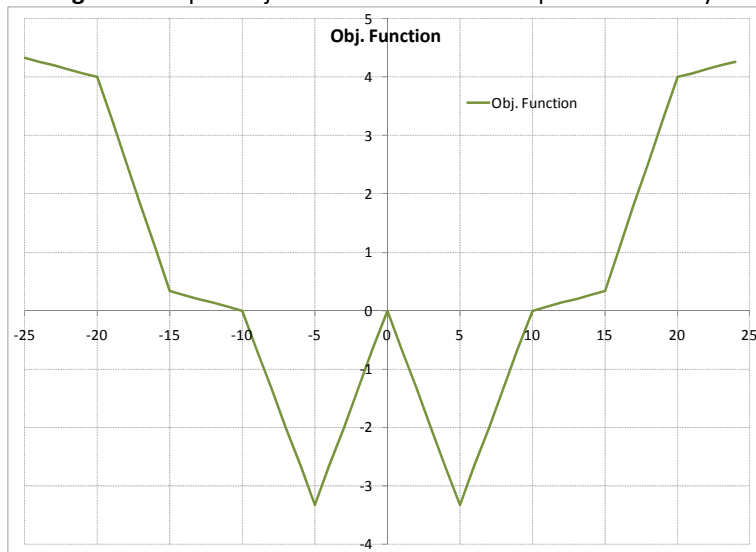


Figure 7. Cross section of objective function through EW direction at the middle of the map.

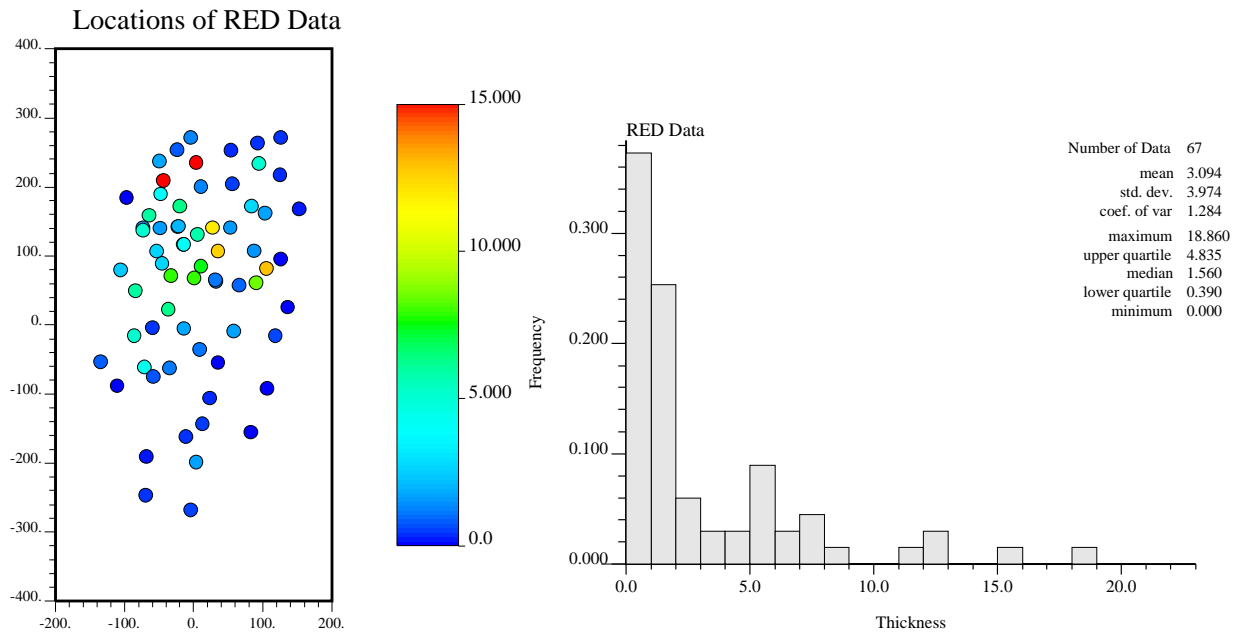


Figure 8. Location map of red data set at left and histogram at right.

```

Parameters for CFD_opt
*****
START OF PARAMETERS:
TR_RED.dat           -file with data
9 10 0              - columns for X,Y,Z
CFD_opt.out          -file for grided obj function
CFD_opt.CFD          -file for Obj>Omax
200 -100 1.0         -nx,xmn,xsiz
200 -100 1.0         -ny,ymn,ysiz
1 0.0 0.0           -nz,zmn,zsiz
1 1.1               -al,a2
45.0001             -alpha
-10                 -Omax
    
```

Figure 9. The parameter file of CFD_opt, which has been used with Red.dat data set.

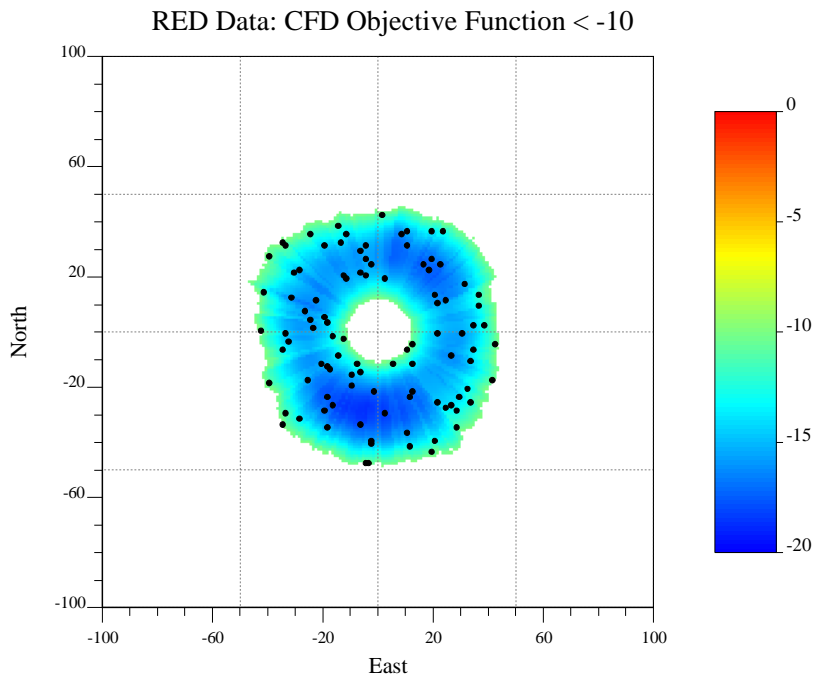
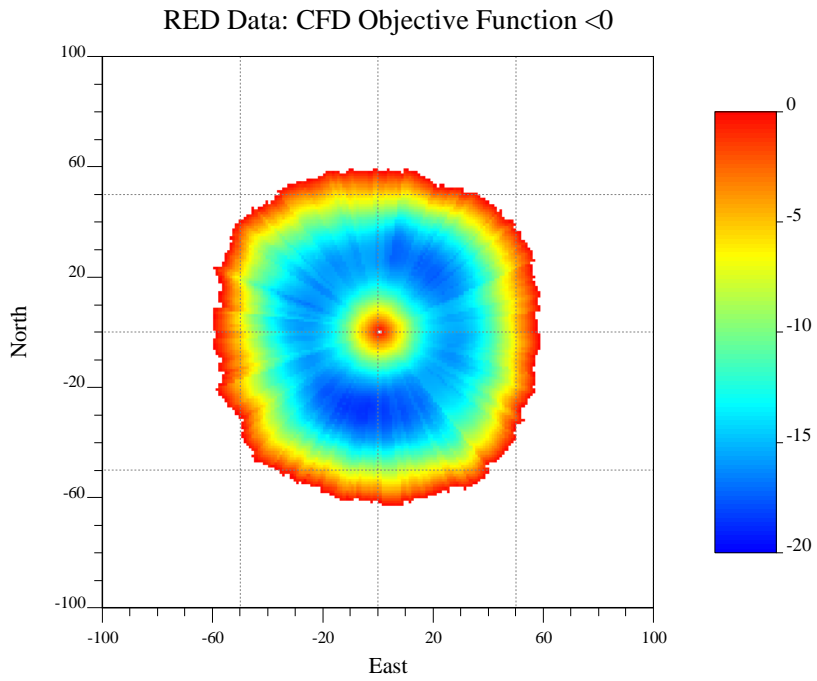


Figure 10. Map of negative values of objective function.

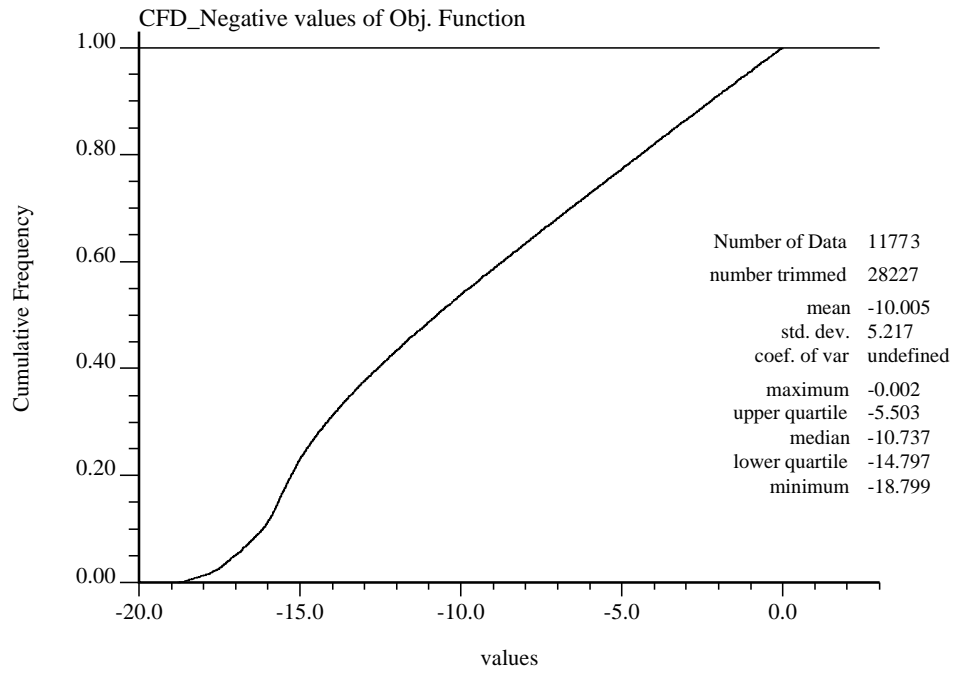


Figure 11. CDF of Negative values of objective function

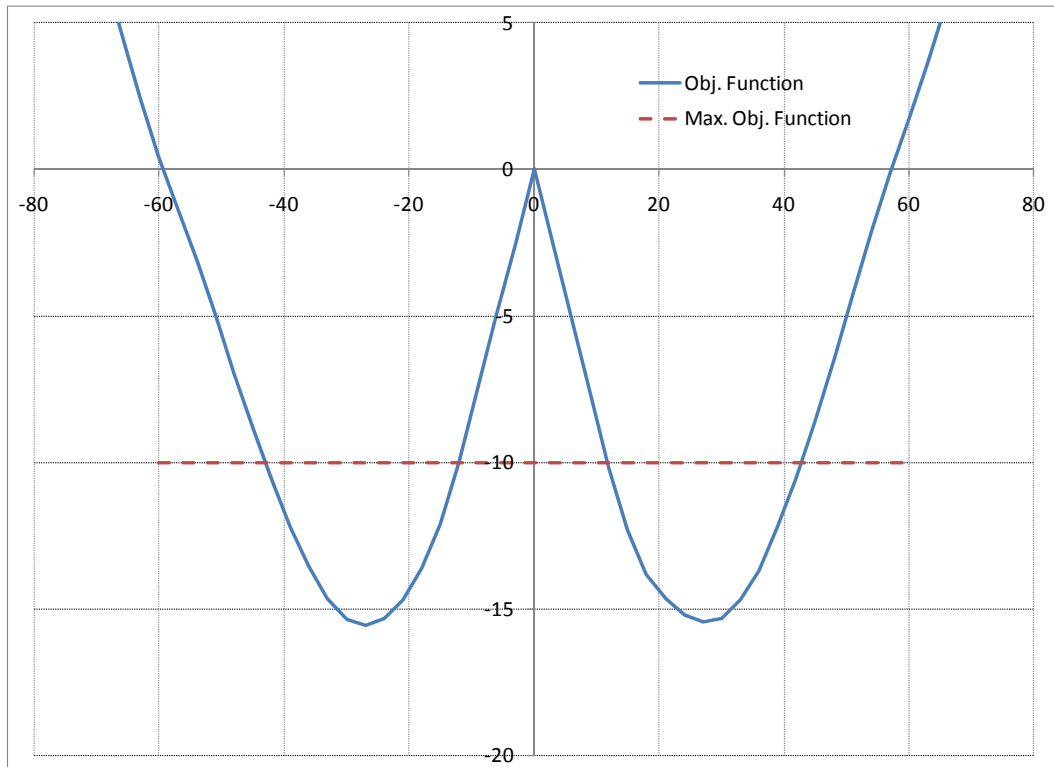


Figure 12. Cross section of objective function through EW direction at the middle of the map for "red" data set.

```

Parameters for SGSIM with Conditional Finite Domain.
*****

START OF PARAMETERS:
TR_RED.dat          -file with data
9 10 0 4 0 0       - columns for X,Y,Z,vr,wt,sec.var.
-100.0             - trimming limits
1 1                -transform the data (0=no,1=yes) CFD.
(0=no,1=yes)
sgsim.trn          - file for output trans table
0                 - consider ref. dist (0=no, 1=yes)
histsmth.out      - file with ref. dist distribution
1 2              - columns for vr and wt
0.0 60.0         - zmin,zmax(tail extrapolation)
1 0.0            - lower tail option, parameter
1 15.0           - upper tail option, parameter
1                -debugging level: 0,1,2,3
sgsim.dbg          -file for debugging output
sgsim.out          -file for simulation output
100               -number of realizations to generate
CFD_opt.CFD       -file with CFD_opt
1 2 3            - columns for X,Y,Z
100              - Number of configuration
6588742          -random number seed
0 20             -min and max original data for sim
20              -number of simulated nodes to use
0               -N/A
1 3             -multiple grid search (0=no, 1=yes),num
0              -maximum data per octant (0=not used)
1000.0 1000.0 10.0 -maximum search radii (hmax,hmin,vert)
0.0 0.0 0.0     -angles for search ellipsoid
51 51 11        -size of covariance lookup table
0 0.60 1.0      -ktype: 0=SK,1=OK
none            - N/A
0              - N/A
1 0.15         -nst, nugget effect
2 0.85 0.0 0.0 0.0 -ivt, cc, angl1, ang2, ang3
250 250 250    -a_hmax, a_hmin, a_vert
    
```

Figure 13. The parameter file of SGS_CFD, which has been used with Red.dat data set.

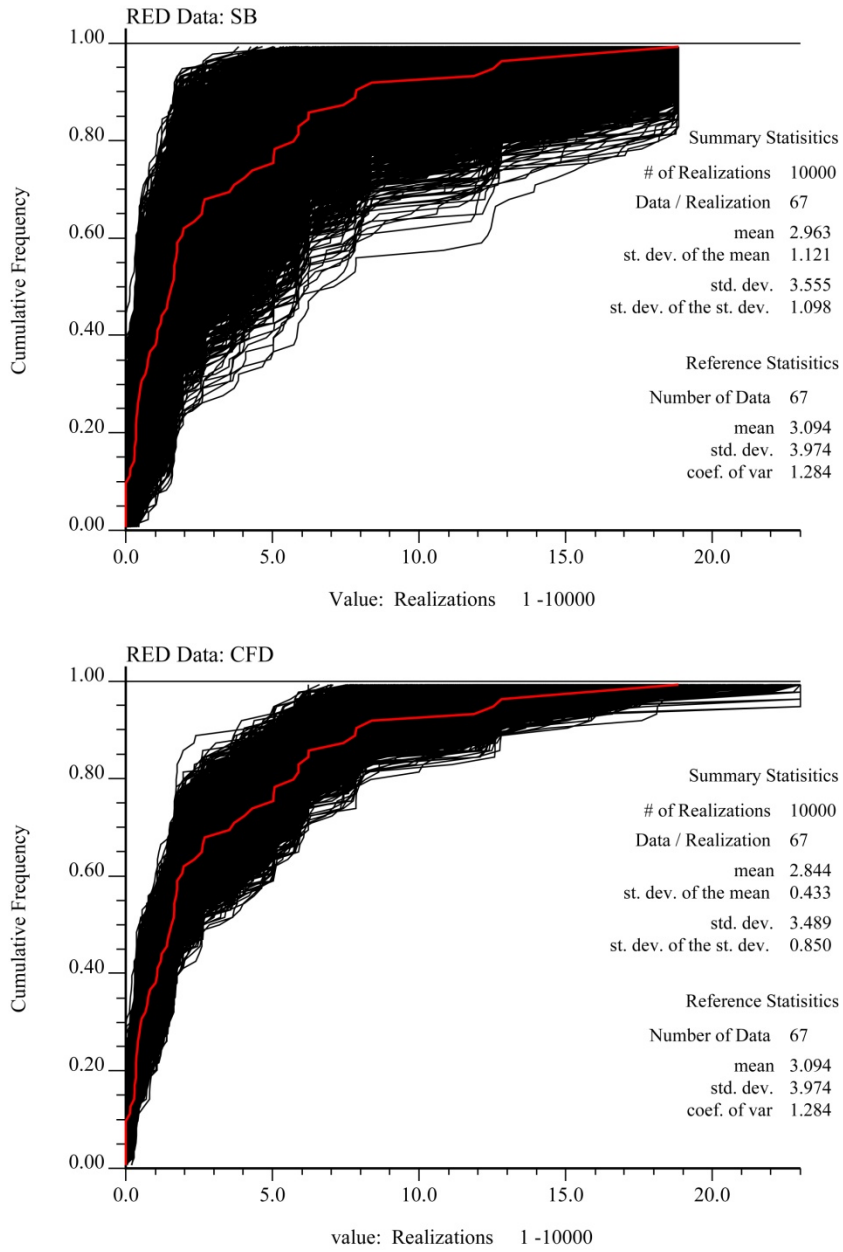


Figure 14. CDF of simulated realization (black lines) with method SB at top and CFD at bottom. Red line is reference distribution

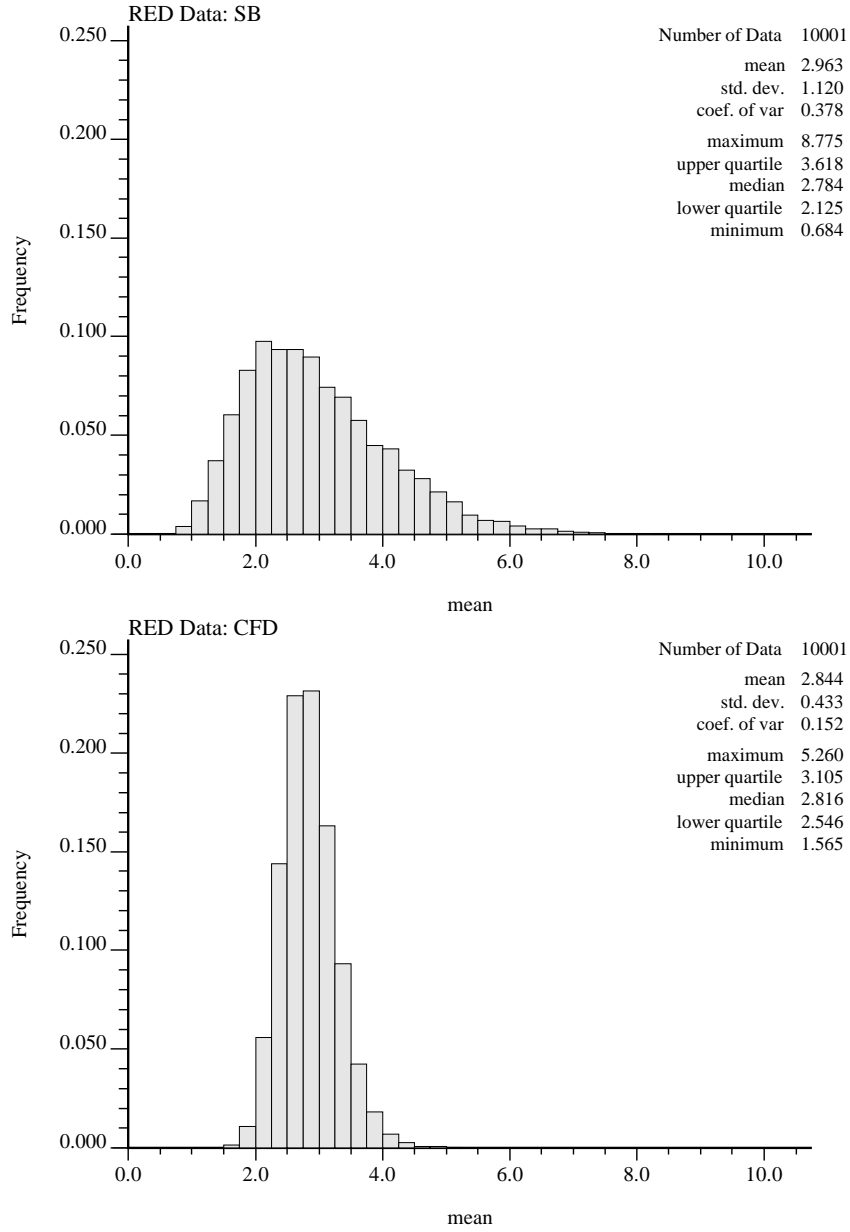


Figure 15. Histogram of mean. SB at top and CFD at bottom.

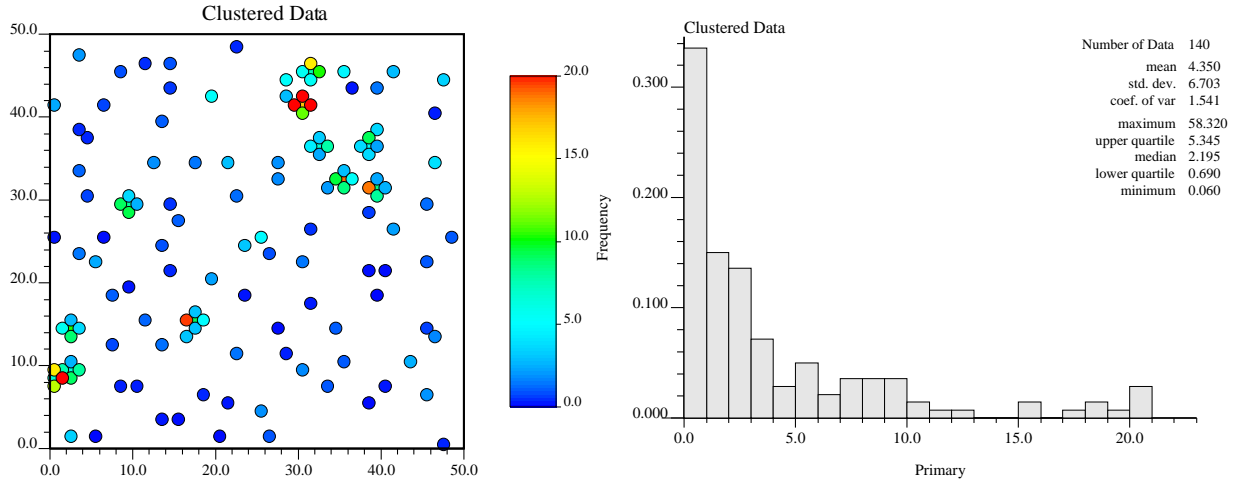


Figure 16. Location map of clustered data set at left and histogram at right.

```

Parameters for CDF_opt
*****
START OF PARAMETERS:
cluster.dat           -file with data
1 2 0                - columns for X,Y,Z
CFD_opt.out          -file for grided obj function
CFD_opt.CFD         -file for Obj>Omax
200 -10 0.1         -nx,xmn,xsiz
200 -10 0.1         -ny,ymn,ysiz
1 0.00 1.0          -nz,zmn,zsiz
1 1.1               -a1,a2
45.0001             -alpha
-0.85              -Omax
    
```

Figure 17. Parameter file of CFD_OPT, which has been used with Clustered data set.

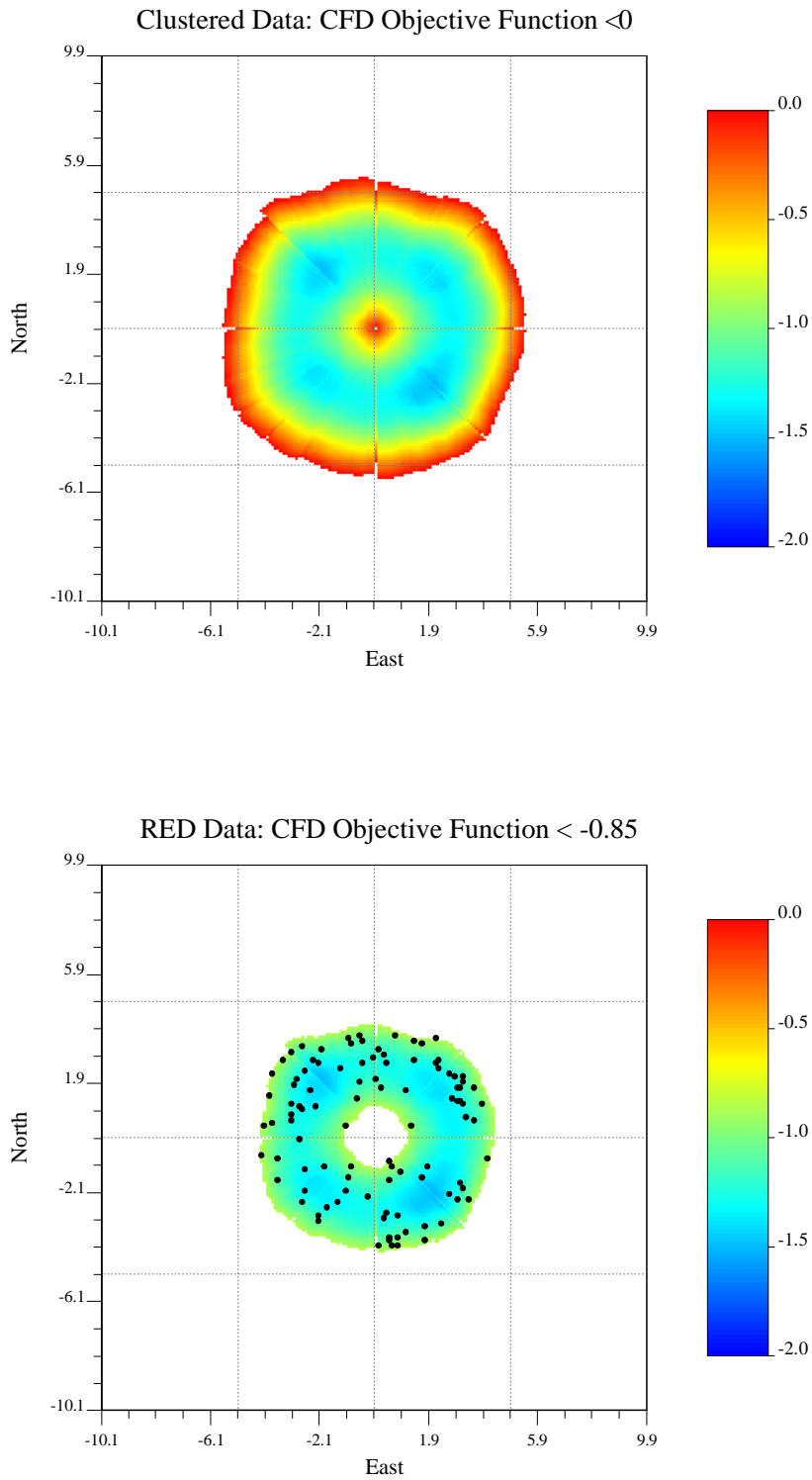


Figure 18. Map of negative values of objective function.

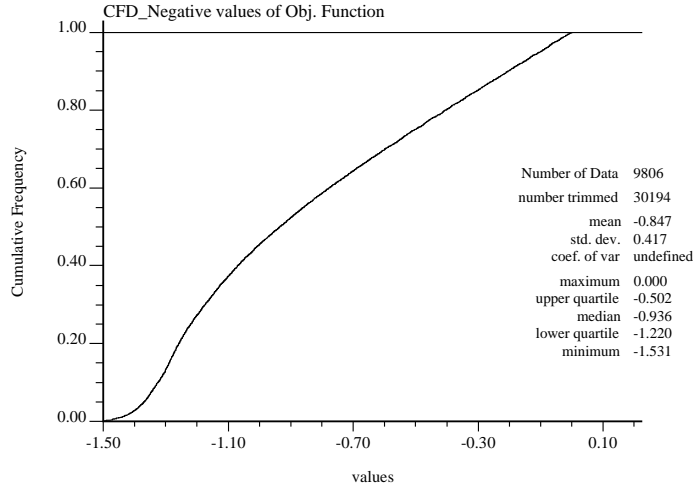


Figure 19. Cumulative distribution of negative values of objective function.

```

Parameters for SGSIM with Conditional Finite Domain.
*****

START OF PARAMETERS:
cluster.dat          -file with data
1 2 0 3 0 0         - columns for X,Y,Z,vr,wt,sec.var.
-100.0             1.0e21    - trimming limits
1 1                -transform the data (0=no,1=yes) CFD.
(0=no,1=yes)
sgsim.trn           - file for output trans table
0                  - consider ref. dist (0=no, 1=yes)
histsmth.out       - file with ref. dist distribution
1 2               - columns for vr and wt
0.0 60.0          - zmin,zmax(tail extrapolation)
1 0.0             - lower tail option, parameter
1 15.0           - upper tail option, parameter
1                -debugging level: 0,1,2,3
sgsim.dbg          -file for debugging output
sgsim.out          -file for simulation output
100               -number of realizations to generate
CFD_opt.CFD       -file with CFD_opt
1 2 3            - columns for X,Y,Z
100              - Number of configuration
69069            -random number seed
0 20             -min and max original data for sim
20              -number of simulated nodes to use
0               -N/A
1 3             -multiple grid search (0=no, 1=yes),num
0              -maximum data per octant (0=not used)
1000.0 1000.0 10.0 -maximum search radii (hmax,hmin,vert)
0.0 0.0 0.0     -angles for search ellipsoid
51 51 11        -size of covariance lookup table
0 0.60 1.0      -ktype: 0=SK,1=OK
none           - N/A
0             - N/A
1 0.2         -nst, nugget effect
1 0.8 0.0 0.0 0.0 -it,cc,ang1,ang2,ang3
    
```

Figure 20. The parameter file of SGS_CFD, which has been used with Clustered data set.

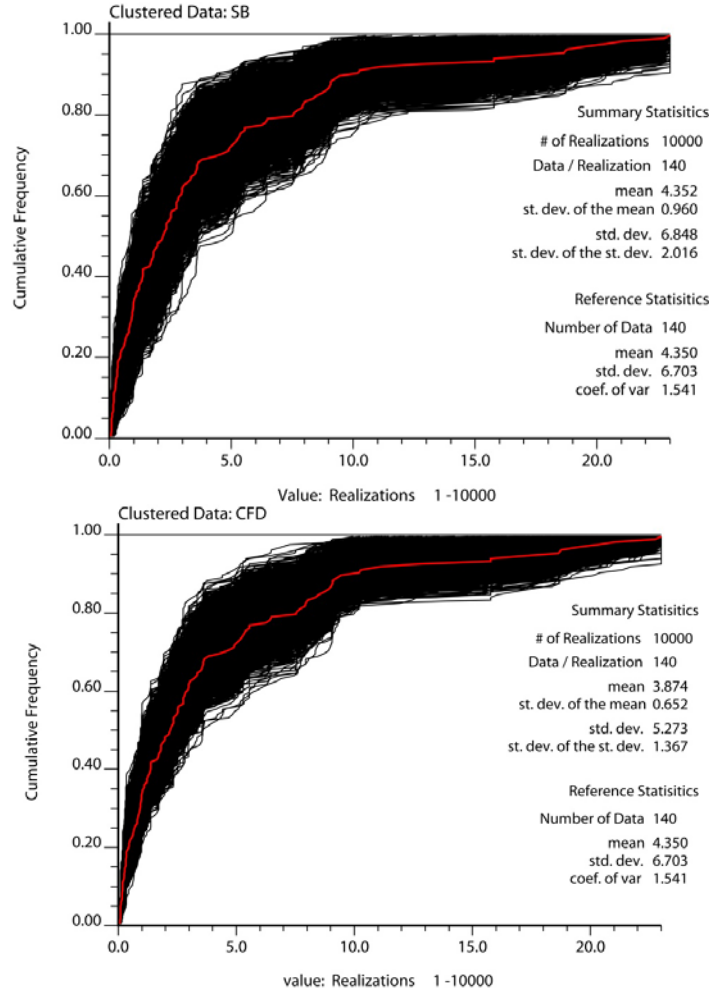


Figure 21. CDF of simulated realization (black lines) with method SB at top and CFD at bottom. Red line is reference distribution

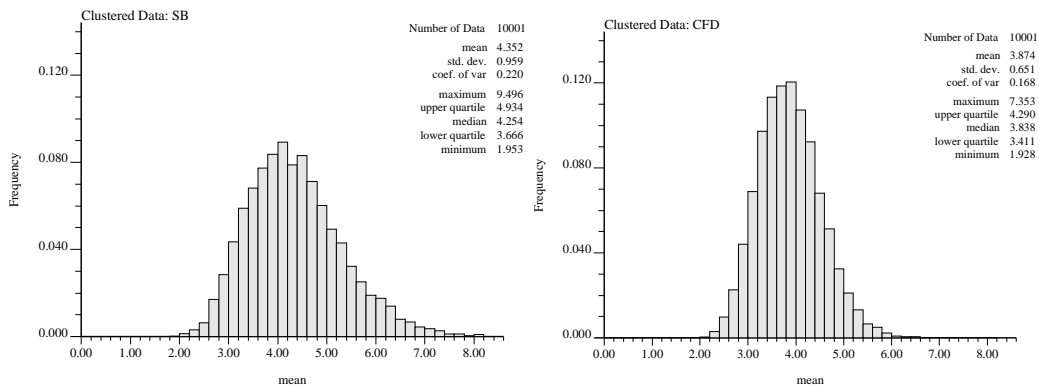


Figure 22. Histogram of mean for Clustered data. SB at top and CFD at bottom.