A New `trans` Program for Histogram and Trend Reproduction

Clayton V. Deutsch

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

Geostatistical simulation is being used increasingly to create alternative realizations that quantify uncertainty. In many cases, the simulated realizations do not reproduce the input target histogram. This is usually considered an advantage: statistical fluctuations are considered an important part of uncertainty. There may, however, be a systematic bias in the reproduction of input statistics In presence of secondary data and non-stationary data. This bias must be corrected. The quantile-quantile transformation procedure enforces histogram reproduction. The procedure can be adapted to respect local data. Further adaptations are described in this short note to enforce vertical and horizontal trends. An iterative approach is required to ensure that local data, the target histogram, and target trends are reproduced.

The Original `trans`

In some applications it is necessary to have a close match between two univariate distributions. For example, a practitioner using a single simulated realization may want its univariate distribution to identify the original data distribution or some target distribution. The GSLIB program `trans` was devised to transform any set of values so that the distribution of the transformed values identifies the target distribution. An important feature of this program is the ability to *freeze* local conditioning data. The transformation is applied gradually as one gets further away from the data locations (Journel and Xu, 1994). When data are frozen, the target distribution is only approximately reproduced; the reproduction is excellent if the number of frozen values is small with regard to the number of values transformed.

Note that fluctuations from model statistics are expected when performing stochastic simulation; it is not advisable to transform all simulated realizations to systematically match the same sample histogram. There can be systematic biases that are important to remove in simulated realizations. These biases may come from (1) variance inflation in cosimulation or simulation with a trend, (2) departures of the data from the implicit multivariate distribution model chosen for the simulation, or (3) implementation considerations such as a limited data neighborhood or numerical precision.

The transformation method is a generalization of the quantile transformation used for normal scores. The p quantile of the original distribution is transformed to the p quantile of the target distribution. This transform preserves the p quantile indicator variograms of the original values. The variogram (standardized by the variance) will also be stable provided that the target distribution is not too different from the initial distribution.

When *freezing* the original data values, the quantile transform is applied progressively as the location gets further away from the set of data locations. The distance measure used is proportional to a kriging variance at the location of the value being transformed. That kriging

variance is zero at the data locations (hence no transformation) and increases away from the data (the transform is increasingly applied). An input kriging variance file must be provided.

Because not all original values are transformed, reproduction of the target histogram is only approximate. A control parameter, ω in [0,1], allows the desired degree of approximation to be achieved at the cost of generating discontinuities around the data locations. The greater ω, the lesser the discontinuities.

This algorithm is generally applied to continuous variables. Categorical values must be ordered. In the case of categorical values a hierarchy or spatial sequencing of the *K* categories is provided implicitly through the integer coding *k=1,...,K* of these categories. Category *k* may be transformed into category *k-1* or *k+1* and only rarely into categories further away. An interesting side application of `trans` is in cleaning noisy simulated images (Schnetzler, 1994). Two successive runs of `trans`, the first changing the original proportions or distribution, the second restituting these original proportions, would clean the original image while preserving data exactitude.

Required Enhancements

The required enhancements could be grouped as (1) accounting for a trend, (2) explicit accounting for uncertainty, and (3) iterative implementation to ensure reproduction of all constraints.

Trends

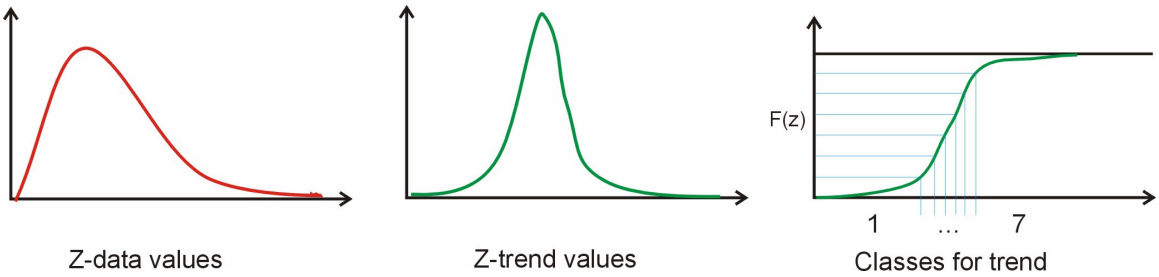
Virtually all geologic variables have a large-scale deterministic trend combined with shorter scale stochastic variations. Geostatistical simulation is aimed at the stochastic variations. The large scale trends are reproduced by geostatistical realizations when there are many data; however, we are often in a case of sparse data. Expert judgment combined with the available sparse data permit inference of a large scale trend model. The trend model provides important local spatial information *and* a representative average that accomplishes declustering/debiasing. The CCG has undertaken extensive research into the inference of trends and their use in geostatistical models. The result of trend modeling is a 3-D trend that represents the expected value at each location:

$$m_{XYZ}(x, y, z)$$

This 3-D trend may have been derived by a moving average, a combination of a vertical and areal trends, or block kriging. Geostatistical realizations are intended to reproduce this trend. The average of the trend over all locations is considered a declustered representative mean. The data histogram must be consistent with this input histogram.

Enforcing reproduction of the trend required *subsetting* the model. If we insist on the trend being reproduced at every grid cell, then we would get back the trend exactly, which is unreasonable. There are three reasonable subsets that we may choose: *nz* vertical trend values, *nx* • *ny* areal trend values, or *nq* classes of trend values based on the input trend model. Choosing to enforce the vertical and areal trend values is reasonable when the 3-D trend was constructed as a combination of these lower-order trends. Considering classes of the trend model may be a better choice with a general 3-D trend model.

The schematic sketch shown below illustrates how $nq=7$ classes would be chosen from a histogram of trend values (the green distributions). The histogram of the data is sketched in red to illustrate how the original data would be more variable.



The approach to enforce reproduction of the trend model in different classes is the same: (1) calculate the target mean in each subset, (2) calculate the actual mean in each subset, and (3) multiply all data in the subset by the ratio of the target divided by the actual. This will be described below with the *Algorithm*.

Explicit Accounting for Uncertainty

Local uncertainty is largely insensitive to large-scale uncertainty in parameters such as the global histogram; however, geostatistical realizations are being used increasingly to assess global uncertainty in recoverable reserves. Techniques such as the spatial bootstrap (see CCG Report Six and other publications) are being used to assess uncertainty in the input histogram or (at least) the input mean. There is a need to account for this input parameter uncertainty. Global uncertainty would be underestimated if the global histogram/trend were imposed on every realization.

Global uncertainty in the univariate distribution should be transferred through to geostatistical simulation and then onto response uncertainty. We could assemble a database of equiprobable (equally likely to be drawn) target univariate distributions by the spatial bootstrap or some similar technique. Each realization could be transformed to reproduce a different target distribution. Practitioners would have to implement this with a script calling the modified `trans` program separately for each realization. A different, simpler, approach is followed in this implementation.

Uncertainty in the histogram shape is considered a second-order effect. Uncertainty in the mean is a first order effect that should be quantified and transferred to each realization. One of the primary output files of the spatial bootstrap program is a file with alternative equiprobable mean values. There is an option to save the full realizations, but we only use the mean values. As an option, this file of different mean values can be input directly to the new `trans` program. The original distribution will be scaled to the mean values in this file.

The original distribution is defined by paired values and weights: (z_i,w_i), $i=1,\ldots,n$. A weighted mean from the original distribution is simply calculated: m_{orig}. There are many ways to scale the data values to a different target mean $m_{tar} \neq m_{orig}$. Note that the local data values are not changed; we only scale the values that constitute the global distribution, which may come from the local data values. A multiplicative approach is followed whereby all data values are transformed by:

$$z_i^t = z_i \bullet \frac{m_{tar}}{m_{orig}}$$

This is applicable to the commonl encountered positively skewed distributions of non-negative variables. It is possible to generate too-large values when $m_{tar} > m_{orig}$. The solution is to reject those values and rescale the remaining data:

$$\text{reject } z_i^t > z_{\max}$$

$$\text{calculate mean of remaining data: } m_t$$

$$\text{scale remaining values: } z_i^{t+1} = z_i^t \bullet \frac{m_{tar}}{m_t}$$

This will have to be applied iteratively. Convergence to a mean within a small tolerance of the target is achieved within five iterations. The new `trans` program will loop this algorithm until the results are within 0.001 of the target mean.

The trend values must also be scaled to be consistent with the target mean. A multiplicative approach is also followed: $m_{XYZ}^t(x,y,z) = m_{XZY}(x,y,z) \bullet \frac{m_{tar}}{m_{trend}}$ where m_{trend} is the mean of the original trend values. The trend values should not be highly variable; otherwise, they are not really a trend. For this reason, no iteration is performed to remove high values.

If this option is chosen, each realization will be scaled to a different target distribution. Randomly pairing an input realization with a target distribution/mean may lead to large changes for some realizations. Ideally, we might chose to transform each realization to a target mean that is close to the original. This has not been implemented.

The Algorithm

The distribution of the initial realization is denoted $F_{init}(z)$. The target distribution is denoted $F_{tar}(z)$. All initial values $z_i, i=1,...,N$ could be transformed to ensure reproduction of the target distribution:

$$z_i^{hist} = F_{tar}^{-1}\left(F_{init}(z_i)\right),\ i = 1,...,N$$

The initial values could also be transformed to ensure reproduction of the trend within the $k=1,...,K$ subsets of the trend model:

$$z_i^{trend} = z_i \bullet \frac{m_{tar}^k}{m_{init}^k},\ i = 1,...,N$$

The k superscript in this equation refers to which class of the mean model the ith data falls. We could attempt to enforce the mean within multiple subsets: e.g., the vertical trend and the horizontal trend. We could add an index for the trend subset: $z_i^{trend,s},\ s = 1,...,N_T$.

The transformed values that would lead to reproduction of the histogram and the trend will likely be different. We could choose to average them to approximately reproduce all of the data constraints:

$$z_i^{ht} = avg(z_i^{hist}, z_i^{trend,s}, s = 1,...,N_T),\ i = 1,...,N$$

The *ht* superscript denotes the transformed values that would approximately reproduce the target histogram and trends. There is still the requirement to reproduce the original local data. The approach of Journel and Xu (implemented in GSLIB) using the kriging variance will be followed:

$$z_i^t = z_i + \left(\frac{\sigma_K^2}{\sigma_{K,\max}^2} \right)^{\omega} \bullet \left(z_i^{ht} - z_i \right), \; i = 1,...,N$$

These final values (with the *t* superscript) will reproduce the local data values exactly and will reproduce the target histogram and trend approximately.

Requirement for Iteration

Ensuring consistency in the target histogram and the trend model may make it possible to enforce the local data, the target histogram and the trend model in one pass; however, in general there will be trade-offs and the transformed values will not exactly reproduce the input statistics. It seems reasonable to reset the initial realization to the transformed realization and rerun the algorithm. This is easy to implement. Experience shows that this permits convergence to all of the objectives provided that they are consistent with each other. Inconsistent inputs such as the target mean being different than the mean of the trend model would lead to the results not converging.

The New Program

The `Trans_Trend` program follows standard GSLIB conventions. Most of the functions are available in GSLIB. Two source code files are required: `Trans_Trend.for` and `Subs.for`; the subroutines have been collected to facilitate compilation if the compiled GSLIB library is not available. The parameters for the program:

```
Line   START OF PARAMETERS:
 1     reference.dat             -file with reference distribution
 2     1   0                     -   columns for variable and weight(0=none)
 3     -1.0e21  1.0e21           -   trimming limits
 4     0.0   75.0                -distribution tails: minimum and maximum value
 5     1      1.0                -   lower tail: option, parameter
 6     1     75.0                -   upper tail: option, parameter
 7     sgsim.out                 -file with original distributions
 8     1                         -   column for variable
 9     50  50  1                 -   nx, ny, nz: size of 3-D model
10     trans.out                 -file for transformed distributions
11     0                         -honor local data? (1=yes, 0=no)
12     kt3d.out                  -   file with estimation variance
13     2                         -   column number
14     0.5                       -   control parameter ( 0.33 < w < 3.0 )
15     1                         -honor trend data? (1=yes, 0=no)
16     trend.out                 -   file with 3-D trend
17     2                         -   column number
18     1 1 0    10               -   subsets: Z, XY, classes of m, number
19     1                         -consider different mean values? (1=yes, 0=no)
20     Spatial_Bootstrap.out     -   file with means
21     1                         -   column number
22     50                        -number of iterations
```

Lines 1-3 specify the input reference distribution. The trimming limits apply to all data files. The tails of the distribution are required when there are relatively few data. **Lines 4-6** specify the tail options. The minimum and maximum are used for all options. Option 1 is linear interpolation, which is recommended. The other options are the same as GSLIB. **Lines 7-8** specify the input realizations that are to be transformed. This program is setup to transform

gridded values: **line 9** specifies the grid size. The output file is specified in **line 10**. Options to condition to local data are specified on **lines 11-14**. The trend is specified on **lines 15-18**. The trend has to be at the same grid resolution as the input models to transform – even though it may have been constructed by lower order trends. The 1/0 indicators specify the subsets that will be considered in enforcing the trend. The number of classes must also be specified on Line 18 – this is used when quantile *classes of m* is used. **Lines 19-21** control the ability to use different mean values for each realization. The input reference distribution and the trend will be scaled to the target mean values read from the file in **line 20** (if turned on). The final option, on **line 22**, controls how many times to loop the program.

This program does not transform categorical variables. There are better approaches to clean categorical variable realizations. This program could be used for the Gaussian deviate that will be used for truncated Gaussian simulation resulting in reproduction of proportions.

GSLIB Example

The small GSLIB data was used for the first example. SGS realizations were created using the `cluster.dat` data with the declustering weights in that data file. Figure 1 shows the results of transforming two realizations to exact histogram reproduction without explicitly reproducing local data. The reference true histogram was used from the file `true.dat`. The Q-Q plot on the lower right shows exact reproduction of the histogram by each realization. The local data were not enforced. Figure 2 data shows the data reproduction; there are minor deviations from the 45^o line. Kriging was performed on the same 50 by 50 grid to get a file with kriging variances, which are required to reproduce local data. The SGS realizations were then transformed to reproduce the histogram and the local data. The local data are reproduced exactly – the results were checked, but no cross plot is shown. Figure 3 shows the realization enforcing local data reproduction – the differences are not visually noticeable. The target histogram is not exactly reproduced with only one iteration (see the left of Figure 4). Five iterations are sufficient to ensure that the target histogram and the local data are reproduced exactly (see the right of Figure 4).

The spatial bootstrap was applied to quantify the uncertainty in the global average, see the left of Figure 5. 5000 realizations were created with the spatial bootstrap. Five hundred SGS realizations were generated with the reference target input histogram. A histogram of the five hundred averages are shown on the right of Figure 5. Notice that the variance of the SGS results is less than 10% of the variance we calculate from the spatial bootstrap. There is a need to explicitly account for the uncertainty in the input histogram (see also the paper by Babak in this CCG report). A reasonable workflow would be to use a different target histogram for every SGS realization. This would require modifying the `sgsim` program or running the program in script. The `Trans_Trend` program could be used with different target mean values. The 5^{th}, 50^{th}, and 95^{th} percentile mean values were calculated from the spatial bootstrap results: 1.634, 2.530 and 3.991. Transforming the first SGS realization to these three mean values leads to the results shown on Figure 6. These maps would be useful for sensitivity analysis: (1) the pattern of spatial variability is the same, (2) local data are exactly reproduced, and (3) the global histogram is different in a controlled fashion – sampling specified quantiles.

Amoco Example

The 3-D Amoco data were used to show trend reproduction. In all cases, the local data were constrained to be reproduced exactly. Figure 7 shows a summary of one SGS realization. The central XY and XZ slices are shown on the left side (the grid is 65 x 65 x 65 = 274,625). The areal average (over Z) and vertical average (over XY) from the first realization are shown on the right side of the figure. The local data enforce a trend even without any explicit trend enforcement in the SGS.

An exaggerated vertical trend going from a low of 2% porosity at the base of the stratigraphic unit to 12% at the top of the stratigraphic unit was imposed in the transformation. The overall average of the vertical trend was rescaled from 7% to 8.33% because of the reference histogram. The reference histogram was taken from the original well data. Summaries of the resulting transformed model are shown on Figure 8. The local data, histogram and vertical trend are reproduced. The model was changed significantly because the vertical trend has significant variability. In fact, many iterations were required to enforce the vertical trend. The vertical trend for the initial SGS realization, 1, 5, 10 and 100 iterations are shown at the bottom of Figure 8.

An areal trend was constructed by block kriging with a 30% nugget effect variogram. This areal trend was enforced by `Trans_Trend`, see the results on Figure 9. The program was set to 10 iterations, which was sufficient to ensure a remarkably close reproduction of the histogram and trend. The data were reproduced exactly by construction.

Reproducing the trend within classes is particularly useful for complex trend models, that is, trends that are not simply in one direction or a particular 2-D plane. A histogram of the trend values from block kriging is shown on the left side of Figure 10. This distribution was subset by 10 thresholds (11 classes) and the trend imposed on the final realization. A cross plot of the 11 subset averages of the final results is shown on the right side of Figure 10. Convergence to reproduction of the histogram and reproduction of a subset trend takes 10 iterations for close reproduction. A default of fifty iterations was adopted after some experimentation. The algorithm is extremely fast; this number of iterations is not a concern.

Application Notes

The `Trans_Trend` program implements a simple algorithm that has widespread applicability. Virtually all geostatistical simulation algorithms are notorious for (1) poorly reproducing a target histogram and trend in presence of non-stationarity and implementation simplifications (e.g., the Markov model), and (2) providing a too narrow range of global uncertainty because limited non-ergodic fluctuations. The `Trans_Trend` program provides an important link in the modeling workflow to closely reproduce target statistics including a reasonable range of uncertainty and complex trends while reproducing local data exactly without discontinuities.

An interesting workflow would be to (1) to generate L geostatistical realizations, (2) rank the L realizations by increasing global average, (3) generate $M>>L$ spatial bootstrap realizations of uncertainty in the global average, (4) rank the M bootstrap averages and extract L equally spaced quantiles, then (5) transform the L realizations to the bootstrap-derived mean values. The CPU effort to generate the final L realizations would not be much more than generating L random realizations. The result is a set of realizations that more fairly samples the space of uncertainty.

Discussion

Geostatistical simulation algorithms inevitably require a strong assumption of stationarity that is inconsistent with most real data. Implementation issues with cosimulation and simulation with a locally varying mean often require an a-posteriori histogram correction. Transforming simulated realizations to reproduce local data, input histograms and input trends is required in many cases. The updated transformation program described in this note permits transformation with multiple objectives, including various representations of trend models. The program is iterative, but runs very fast even for large 3-D models.

The ability to transform to any target mean value is useful. The uncertainty in the mean can be established with the spatial bootstrap, and then realizations can be transformed to selected quantiles, (e.g., the 0.05, 0.5, and 0.95 quantiles) for sensitivity analysis.

There will be artifacts and unreasonable spatial features if the objectives (data, target histogram or trends) are inconsistent with the initial realization being transformed. Uncertainty will be underestimated if all realizations are transformed to the same distribution. Categorical variables are not handled by this program; an image cleaning program is recommended for that purpose.

References

- Deutsch, C.V. and Journel, A.G., 1998. *GSLIB: Geostatistical Software Library: and User's Guide*. Oxford University Press, New York, 2nd Ed.
- Journel, A.G. and W. Xu, 1994, Posterior Identification of Histograms Conditional to Local Data, Math Geology, 22(8), pp 323-359.
- Schnetzler, E., 1994, Visualization and Cleaning of Pixel-Based Images, M.Sc. Thesis, Stanford University, Stanford CA.

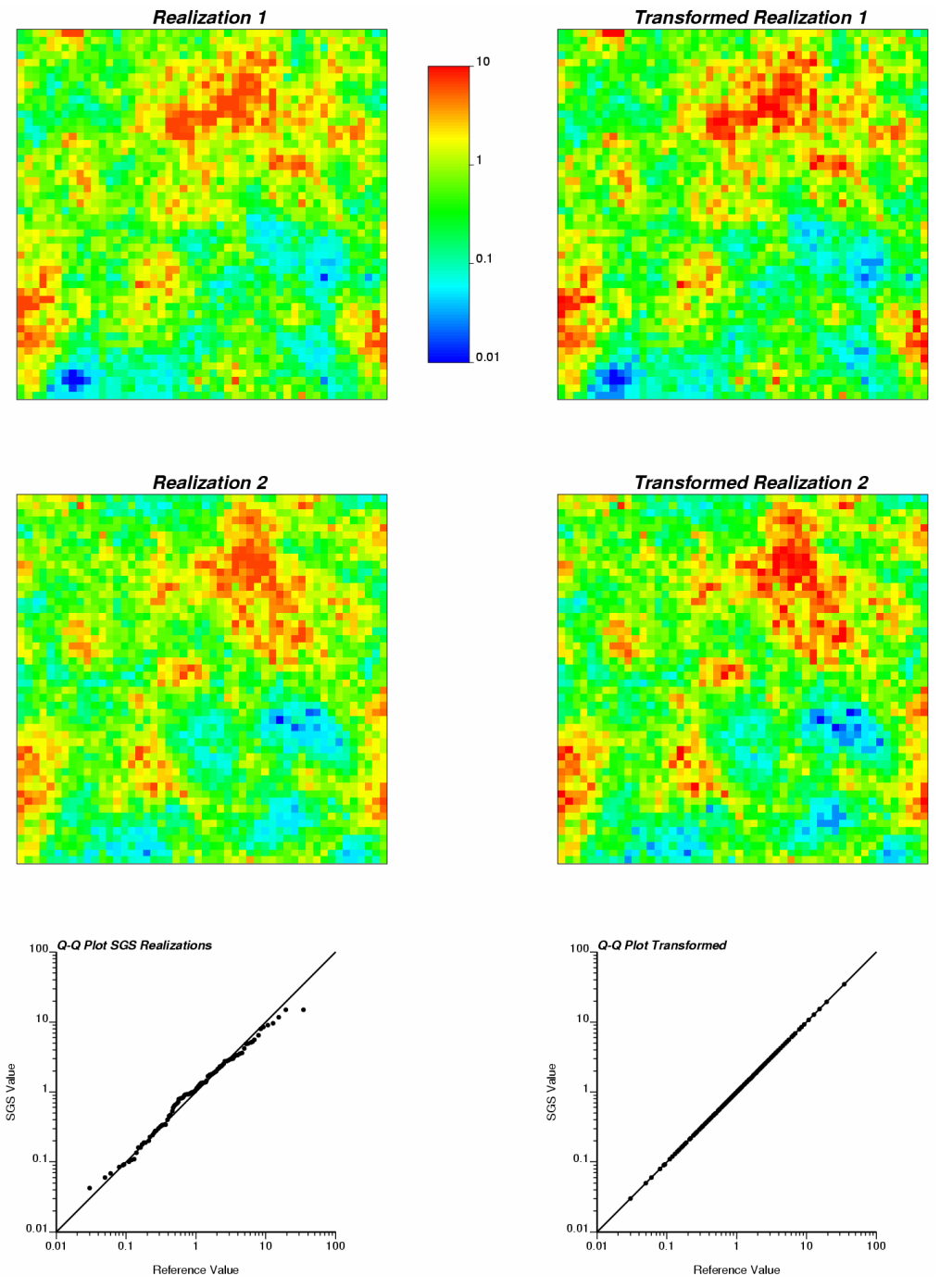


Figure 1: transforming two realizations to exact histogram reproduction without explicitly reproducing local data.

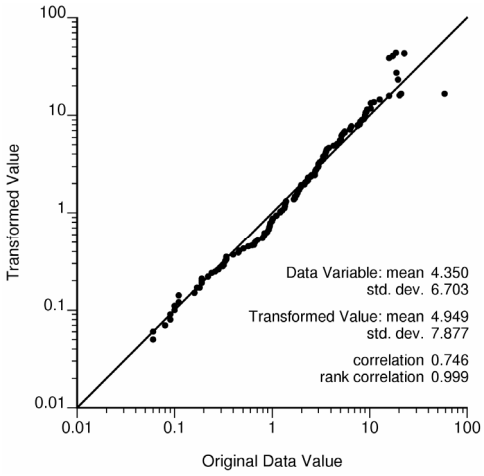


Figure 2: data reproduction and transforming a realization to exact histogram reproduction (without explicitly reproducing local data).

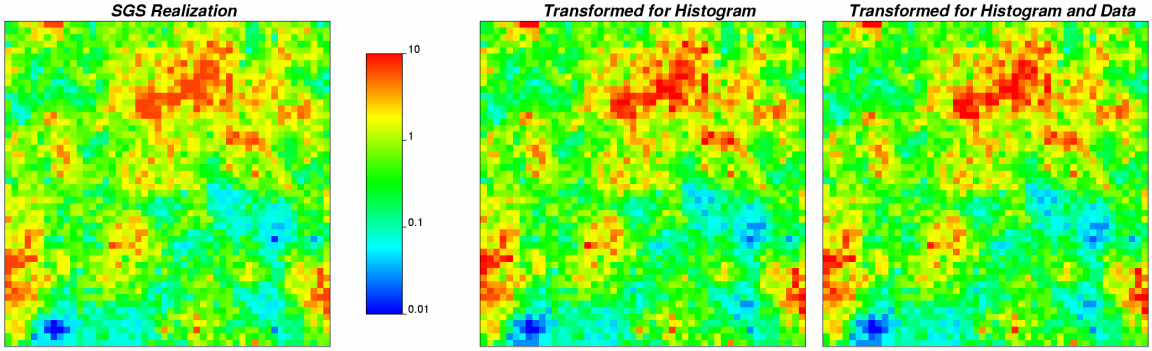


Figure 3: the first realization, reproducing the histogram only, and the histogram plus local data.

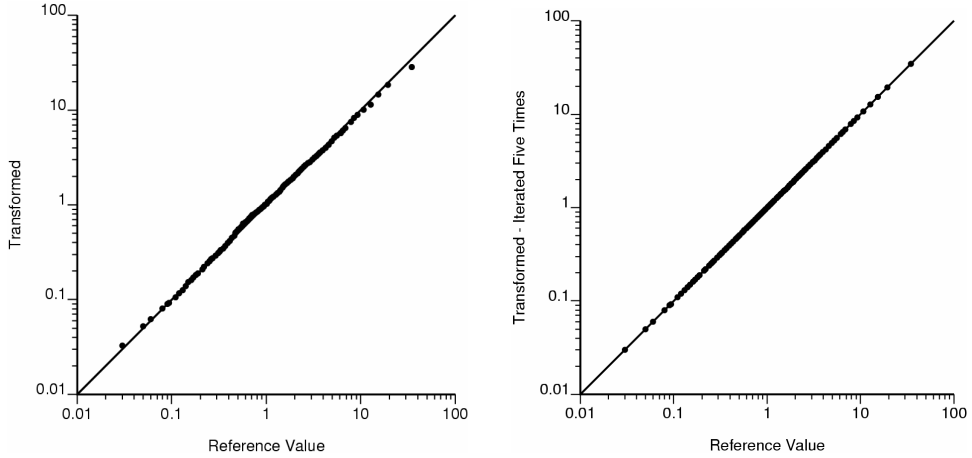


Figure 4: histogram reproduction when local data reproduction is enforced. The plot on the left shows the results after one iteration; the plot on the right shows the results after five.

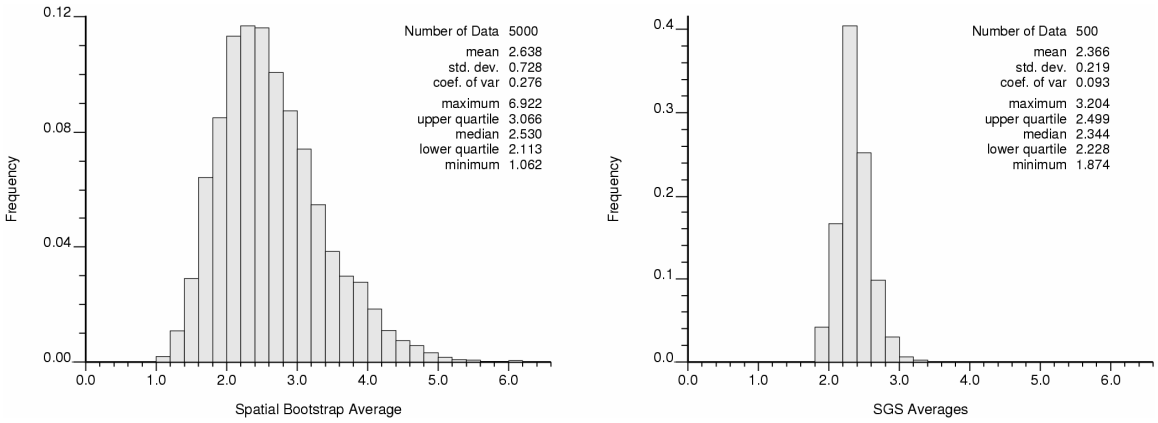


Figure 5: uncertainty in the average predicted by the spatial boostrap (left) and the variability observed in 500 SGS realizations. The variance of the SGS realizations is 9% of that predicted by the spatial bootstrap.

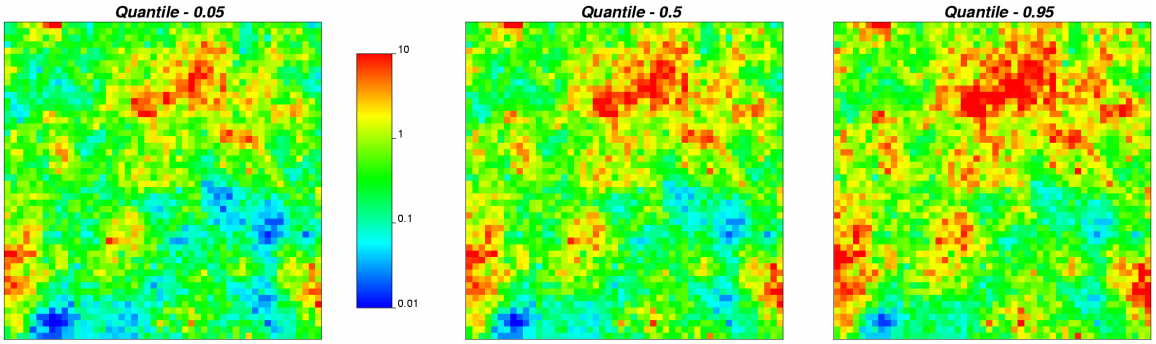


Figure 6: An SGS realization transformed to the 5th, 50th, and 95th percentile uncertainty based on the distribution of uncertainty from the spatial bootstrap. Local data are reproduced.

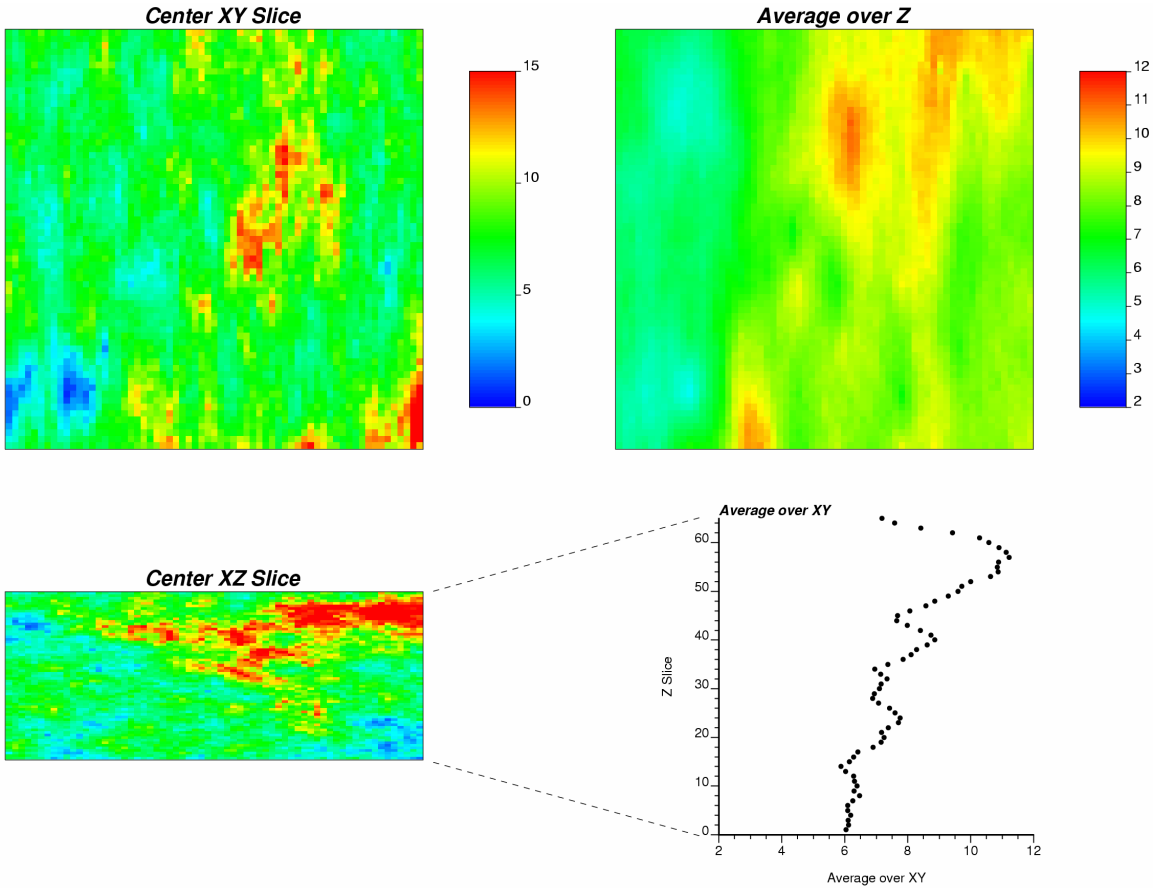


Figure 7: Summary of an SGS realization using the 3-D Amoco dataset (two slices on the left and trend summaries on the right). The realization is conditional to 62 wells, which enforces some areal and vertical trend.

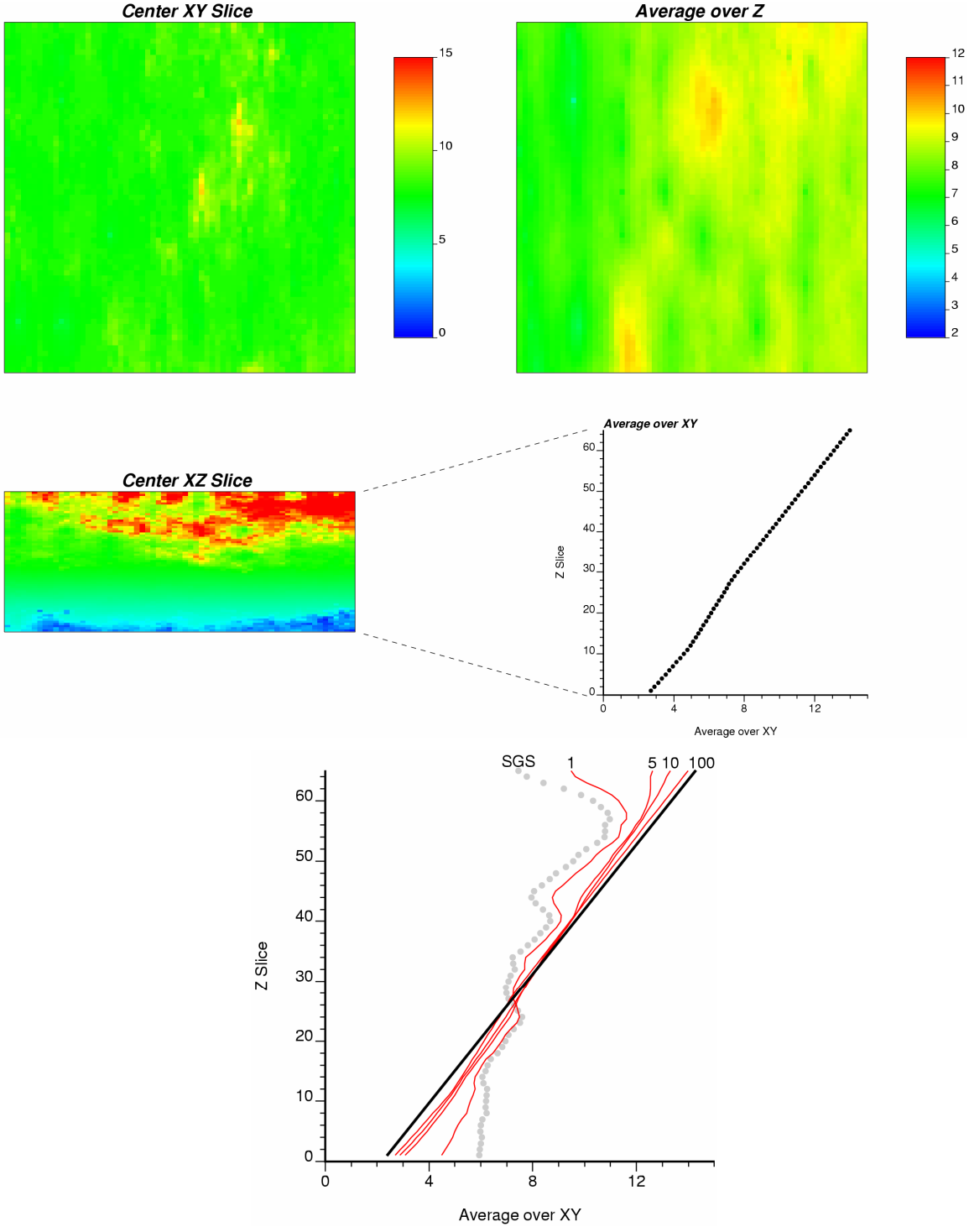


Figure 8: Summary of an transformed results with a strong synthetic vertical trend. The upper four figures are the same summaries as Figure 7. The lower chart shows convergence of the vertical trend with increasing numbers of iterations.

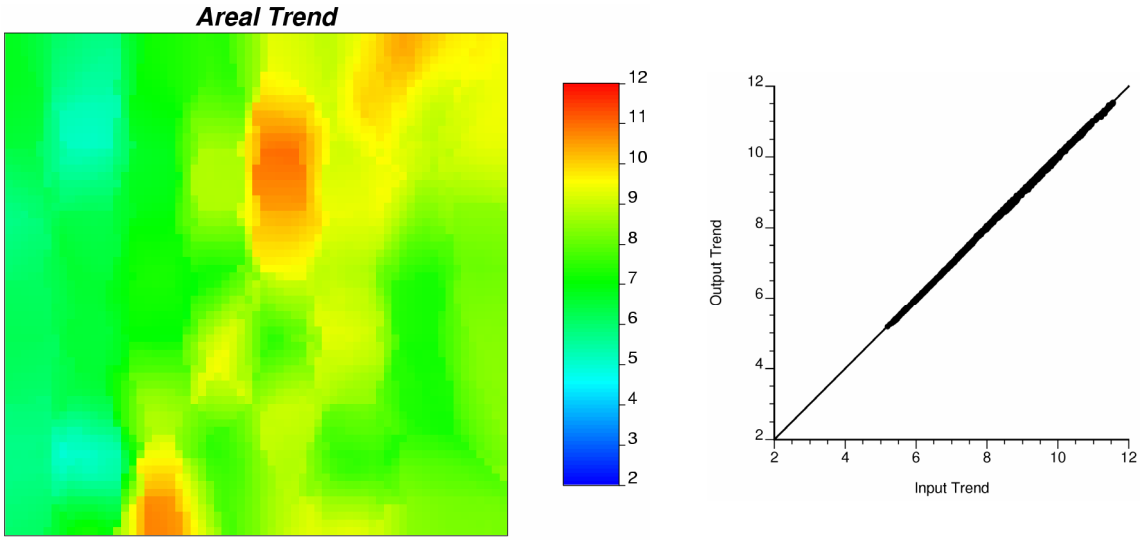


Figure 9: Input areal trend and the reproduction of the areal trend after `Trans_Trend`. Local data and the input histogram were also reproduced.

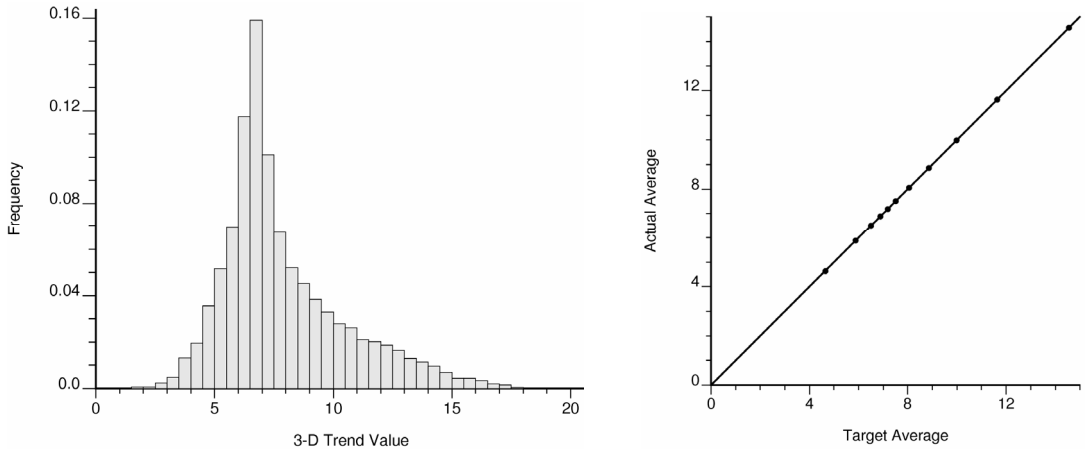


Figure 10: Example of imposing quantiles of a trend model. The histogram of trend values is shown on the left and the reproduction of the trend within each class is shown on the right.