Direct Sequential Simulation of Tartan Grids with Nested Radial Grids

M. J. Pyrcz (mpyrcz@ualberta.ca) and C. V. Deutsch (cdeutsch@ualberta.ca)
University of Alberta, Edmonton, Alberta, CANADA

Abstract

Direct sequential simulation (DSS) does not require a nonlinear Gaussian transformation. This greatly simplifies the integration of data of different support sizes. Original data and previously simulated nodes may be at different volume supports. A FORTRAN program, TARTANSIM, is presented which performs simulation directly to tartan grids. A module is included to insert locally refined radial grids. The program is based on the building block subroutines documented earlier in this CCG Report. ***.*** *Results are discussed and validated by comparison to the theoretically established block averaged sequential Gaussian simulation. A major limitation of the current algorithm is that the results are homoscedastic by construction. SGSIM also works within a homoscedastic framework, but the a priori Gaussian transform and posteriori back transform preserve general heteroscedastic features from the data. This issue is being addressed.*

Introduction

A common methodology for building stochastic models on irregular grids has been to simulate on a fine grid and then to average the local realizations to a coarser irregular grid. This method is computationally expensive. A FORTRAN program which directly simulates to tartan and nested radial grids, called TARTANSIM, is presented in this paper. The general program flow is described, and examples are validation by comparison to theoretically established SGSIM. The current program is limited to a homoscedastic framework. This program is a major step in the direction of directly simulating to general irregular grids.

The Building Blocks

Detailed information concerning the generic subroutines within TARTANSIM are presented in the paper *Building Blocks for Direct Sequential Simulation on Unstructured Grids* (Pyrcz and Deutsch, 2002), which is included in this CCG report. For greater detail into the building block subroutines the reader is referred to this source.

In addition to these general building blocks, there are specific routines for tartan and radial grids such as and building nonstationary covariance tables and kriging matricies. A modified PIXELPLT program from GSLIB (Deutsch and Journel, 1998), called TARTANPIX is included that helps visualize the resulting tartan and nested radial grid simulations. All illustrations of TARTANSIM output were generated with this program.

The Parameters

The algorithm is based on the F90 SGSIM code (Deutsch and Journel, 1998). The general flow, parameter file format and annotation is the same. The parameter file is shown in Figure 1. The new parameters are indicated in bold font.

The *CBAR* parameters are applied by the PREGBAR and CALCGBAR subroutines that calculate the required mean covariances. The *randomsamp* parameter determines whether random samples are applied to calculate the mean covariance. If set to 0, then the traditional descritization method is applied. The *ndis* parameter is the number of random samples or the number of discretizations and *nl* is the number discritizations applied to the variogram model. The *CCDF* parameters are applied in the PRECDIST program to build the CCDF lookup table for the purpose of accessing viable conditional local distributions. The first three parameters, *nm*, *nv*, and *nq* indicate the number of descritizations used in the CCDF lookup table and the parameters *limmean*, and *limstd* control the limits of the descritization in Gaussian space.

The grid definition applied in TARTANSIM is demonstrated in Figure 3. Unlike traditional GSLIB the origin is set at the lower, bottom left corner of the data space. Also, note that for true 2D realizations the thickness in the z direction should be set to 0.0.

The Program Flow

The data to be used for the construction of the CCDF table is read. This data may either by the actual conditioning data or a reference distribution. The PRECDIST subroutine is called to build the CCDF look up table. Unconditional realizations must have a reference distribution.

The PREGBAR subroutine is called to build the geometric search matrices and the variogram lookup table. This allows the subsequent mean covariance calculations to lookup the variogram values as oppose to recalculating. For accurate results many (>10,000) bins should be applied to the variogram model.

A subroutine unique to TARTANSIM, called TARTANINTERPRET, is called next. It translates the tartan grid parameters into a table with block information for each node in the model. This information includes the extents in x,y,z directions and the center location. These parameters are illustrated in Figure 3.

Next the nonstationary covariance table is constructed. This subroutine is described in the building blocks paper, but there has been some significant changes made to the subroutine. The previous implementation was based on building a nonredundant table of the mean covariances between all model blocks and conditioning data. This table was sorted in the order of decreasing mean covariance and the table would be searched from the start for each mean covariance. This was inefficient given the generally large size of the table (table size is a function of grid size and the search ranges).

The new method is to build a table which is ordered by a i index where i = 1, number of nodes and then by a j index where j = 1, number of nodes. Due to redundancy (mean covariance between block i and j and mean covariance between block j and i are included in the table) this increases the size of the table and the number of initial mean covariance calculations, but this method allows for the use of an index look up. For each i index all the j entries are sorted in the order of increasing mean covariance, and the table location of each of these subsets is recorded in a separate array. To retrieve the most correlated neighbouring nodes, no searching of the table is required. While the start-up time is slightly longer than the previous nonredundant method and memory requirements are higher, the run time is much faster (the increase in speed is a function of the table size).

During simulation and at each location the MATRIXBUILD subroutine then searches the nonstationary covariance tables and builds the kriging matrix in the format required by the standard GSLIB matrix solution subroutine, KSOL (Deutsch and Journel, 1998). If there any weights above a threshold (hard coded as +/-1.5) in the example then the problem data is removed and the kriging is repeated. This data is generally screened and this results in the large problematic weights.

An local conditioned distribution is calculated with the GETCDIST subroutine given the calculated kriging mean and variance and Monte Carlo drawing is applied as usual. In this manner all nodes are visited.

The Speed Advantage of TARTANSIM

All of the heavy computational steps are completed prior to simulation. The CCDF lookup table is calculated and indexed for a super block search, and all the mean covariances are stored in indexed tables. This computational effort required for this initial setup is a function of the grid size, the search range, and the level of descritization applied to the CCDF table and the mean covariance calculations. During simulation the only significant calculation is the solving of the kriging matrix. In the following example, the 201 realizations of block average SGSIM required about 75 seconds while the TARTANSIM equivalent required 20 seconds. This advantage becomes even more pronounced when the number of simulations is increase. For example, 1001 realizations would require about 1 minute for TARTANSIM and about 6 minutes for SGSIM.

Unconditional Example

Unconditional simulations generated by SGSIM and TARTANSIM were compared (see Figure 4 and 5). The SGSIM realizations were simulated on a 100 x 100 grid and then block averaged to a 15 x 15 tartan grid. The variogram was set as a single isotropic spherical structure with a range equal to 1/5 of the size of the model. The e-type estimates over 201 realizations were compared (see Figure 6). These e-type estimate maps reveal that TARTANSIM results are unbiased. Some local distributions of uncertainty for both TARTANSIM and SGSIM are compared in Figure 7. These local distributions of uncertainty are very similar. The covariance between adjacent blocks was checked. The actual mean covariance calculated with the variogram model was compared to the empirical covariance between adjacent blocks over 201 realizations. Also, the empirical covariance of TARNSIM and SGSIM were compared (see Figure 8). In this example the covariance is marginally higher in the TARTANSIM realizations, although the results are similar and in general the actual covariance is reproduced.

Conditioned Example

The red 2D data set was applied as conditioning. The Gaussian transformation was applied to remove heteroscedasticity and no declustering was applied. See Figure 9 for the distribution and location map of the transformed data set.

201 realizations where generated with SGSIM with 100 x 100 cell. These realizations were block averaged to a tartan grid (see Figure 10 for some example realizations). The same conditioning

data were applied to TARTANSIM and the simulation was performed directly to the tartan grid (see Figure 11 for some example realizations).

The e-type estimate was calculated over the 201 realizations for both TARTANSIM and block averaged SGSIM results (see Figure 12). The e-type maps indicate that the conditional means from TARTANSIM are similar to the theoretically established SGSIM results. Some example local distributions of uncertainty were compared (see Figure 13). The local distributions of uncertainty generated with TARTANSIM are very similar to the SGSIM distributions. The covariance between adjacent blocks was once again checked (see Figure 14). With the presence of a high degree of conditioning the empirical covariance is systematically underestimated. The TARTANSIM results closely agree with the SGSIM results.

Radial Nested Grids

A module is included with TARTANSIM for the simulation of nested radial grids. The gird parameters and format are illustrated in Figure 15. Activation of this module requires the addition of a line at the end of the parameter file with a 1, to switch radial grids on, the number of radial grids and the list of indexes at which the radial grids will be placed. The radial grids are directly simulated after the entire tartan grid has been simulated. Only the simulated value for the coincidental tartan grid block is retained as conditioning. The method for simulation is similar to that of TARTANSIM. An adapted nonstationary covariance table subroutine called RADIALNONSTATCOVTABLE, and a matrix building subroutine called RADIALMATRIXBUILD is used. Since all the mean covariances are precalculated and stored in a 2D matrix the search and construction of the kriging matrices is very rapid. An example with the previous conditioned data set and 6 nested radial grids is shown in Figure 16.

Conclusion

This work has demonstrated the viability and efficiency of direct simulation to irregular grids. The DSS with nonstationary covariance tables and CCDF look up tables has been shown to be robust, and to calculate results comparable to the theoretically established SGSIM method. As methods are developed to input the proportional effect this code will be updated. Although, there is not quantitative assessment of the severity of ignoring the often present proportional effect, this limitation makes this the current code not appropriate in simulating actual geologic settings.

Acknowledgements

We are grateful to the industry sponsors of the Centre for Computational Geostatistics at the University of Alberta and to NSERC and ICORE for supporting this research.

References

- Deutsch, C.V. and Journel A.G. (1998). *GSLIB: Geostatistical Software Library: and User's Guide*, Oxford University Press, New York, 2nd Ed.
- Pyrcz, M.J. and Deutsch, C.V. (2002). *Building Blocks for Direct Sequential Simulation on Unstructured Grids*, Centre for Computational Geostatistics 2nd Annual Report, University of Alberta.

Parameters for TARTANSIM
****

```
START OF PARAMETERS:
data.dat                              -file with data
1  2  0  6  0  0                      -  columns for X,Y,Z,vr,wt,sec.var.
-100.0       1.0e21                   -  trimming limits
1 500 10000                           -CBAR: random samples (1-yes,0-discr.), ndis, nl
100 100 100 3.0 1.5                   -CCDF Table: nm, nv, nq, limmean, limstd
0                                     -consider ref. dist (0=no, 1=yes)
refdist.dat                           -  file with ref. dist distribution
1  0                                  -  columns for vr and wt
-3.0    3.0                           -  zmin,zmax(tail extrapolation)
1       -3.0                          -  lower tail option, parameter
1        3.0                          -  upper tail option, parameter
0                                     -debugging level: 0,1,2,3
tartansim.dbg                         -file for debugging output
tartansim.out                         -file for tartan simulation output
201                                   -number of realizations to generate
15 0.0 6 5 4 3 2 2 2 2 2 2 2 3 4 5 6  -tnx, originx, dx()
15 0.0 6 5 4 3 2 2 2 2 2 2 2 3 4 5 6  -tny, originy, dy()
1 0.5 1                               -tnz, originz, dz()
69069                                 -random number seed
0     8                               -min and max original data for sim
12                                    -number of simulated nodes to use
1     3                               -multiple grid search (0=no, 1=yes),num
0                                     -maximum data per octant (0=not used)
20   20   1                           -maximum search radii (hmax,hmin,vert)
0.0    0.0    0.0                     -angles for search ellipsoid
0   0.0   1.0                         -ktype: 0=SK,1=OK,2=LVM,3=EXDR,4=COLC
../data/cluster.dat                   -  file with LVM, EXDR, or COLC variable
1                                     -  column for secondary variable
1    0.0                              -nst, nugget effect
1    1.0  0.0   0.0   0.0             -it,cc,ang1,ang2,ang3
         10.0   10.0  10.0            -a_hmax, a_hmin, a_vert
```

Figure 1: Example TARTANSIM parameter file.

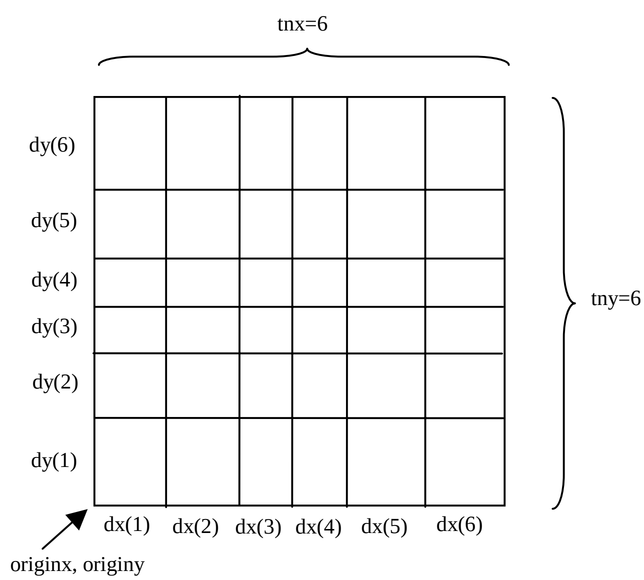

Figure 2: An illustration of the grid parameterization applied in TARTANSIM.

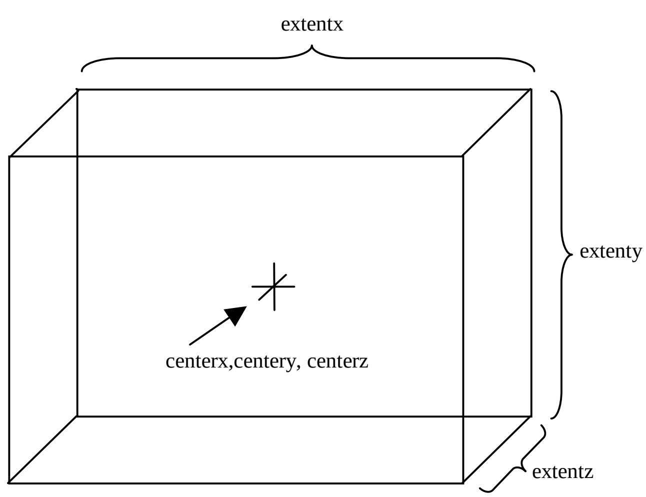

Figure 3: The block parameters stored by the TARTANINTERPRET subroutine.

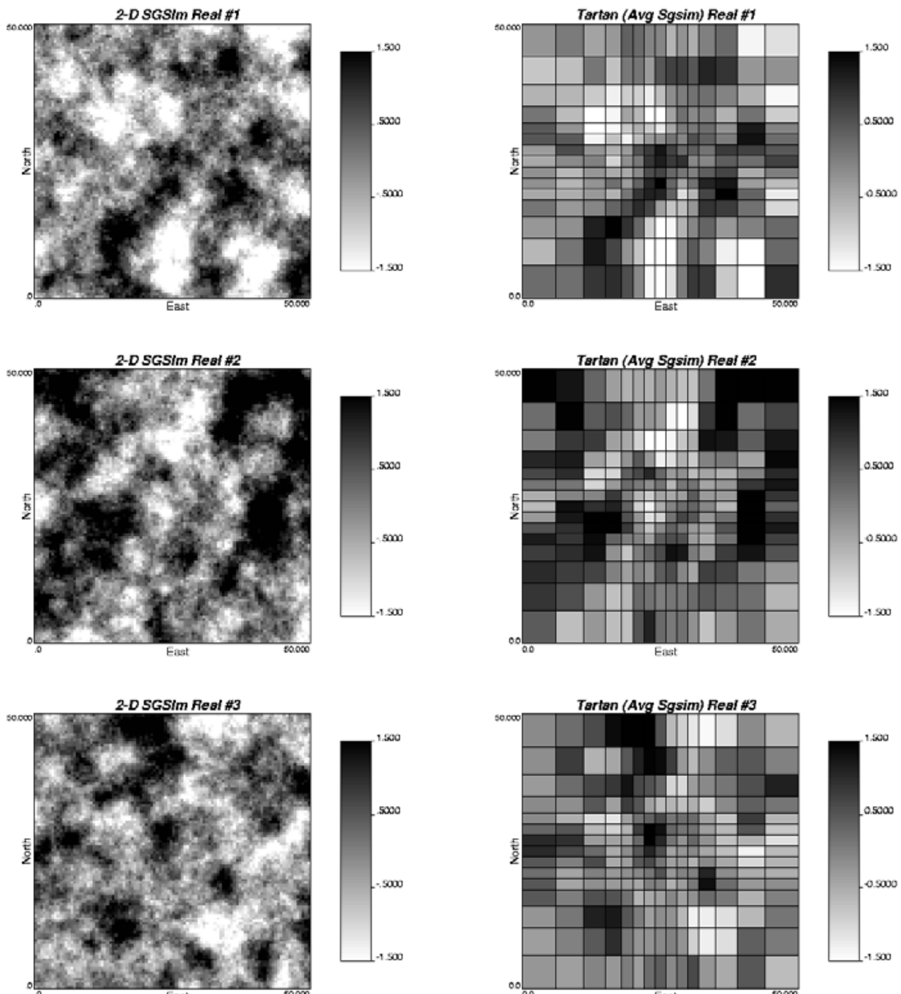

Figure 4: Example unconditional SGSIM realizations and the associated block averaged results.

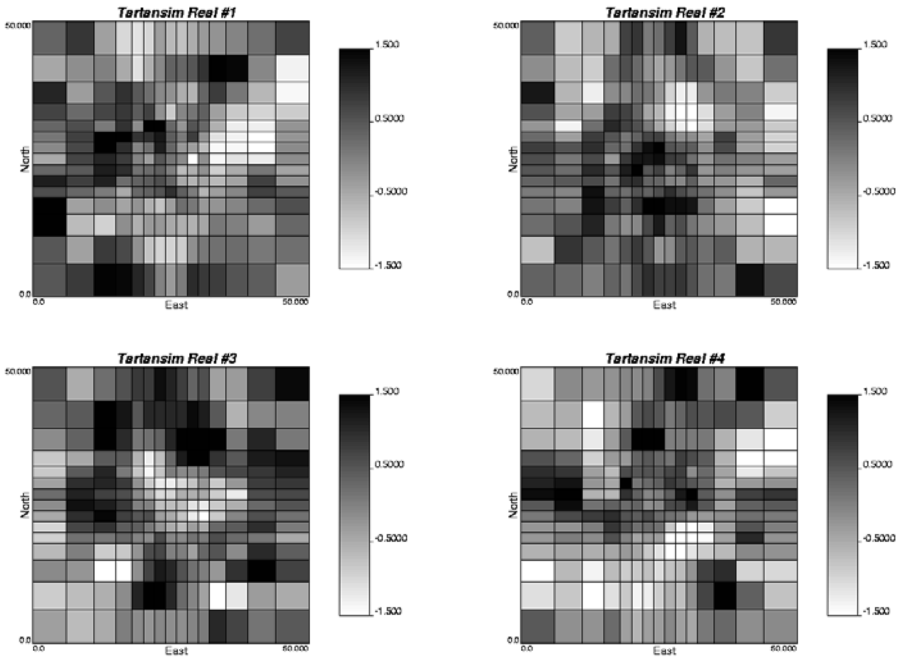

Figure 5: Example unconditional TARTANSIM realizations.

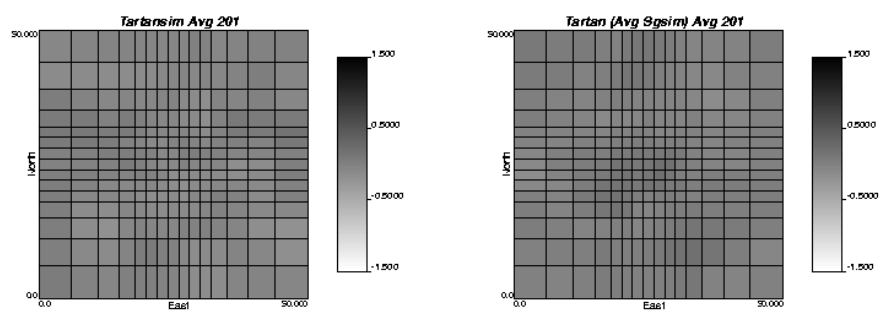

Figure 6: E-type estimates based on 201 realizations of TARTANSIM (left) and SGSIM block averaged (right).

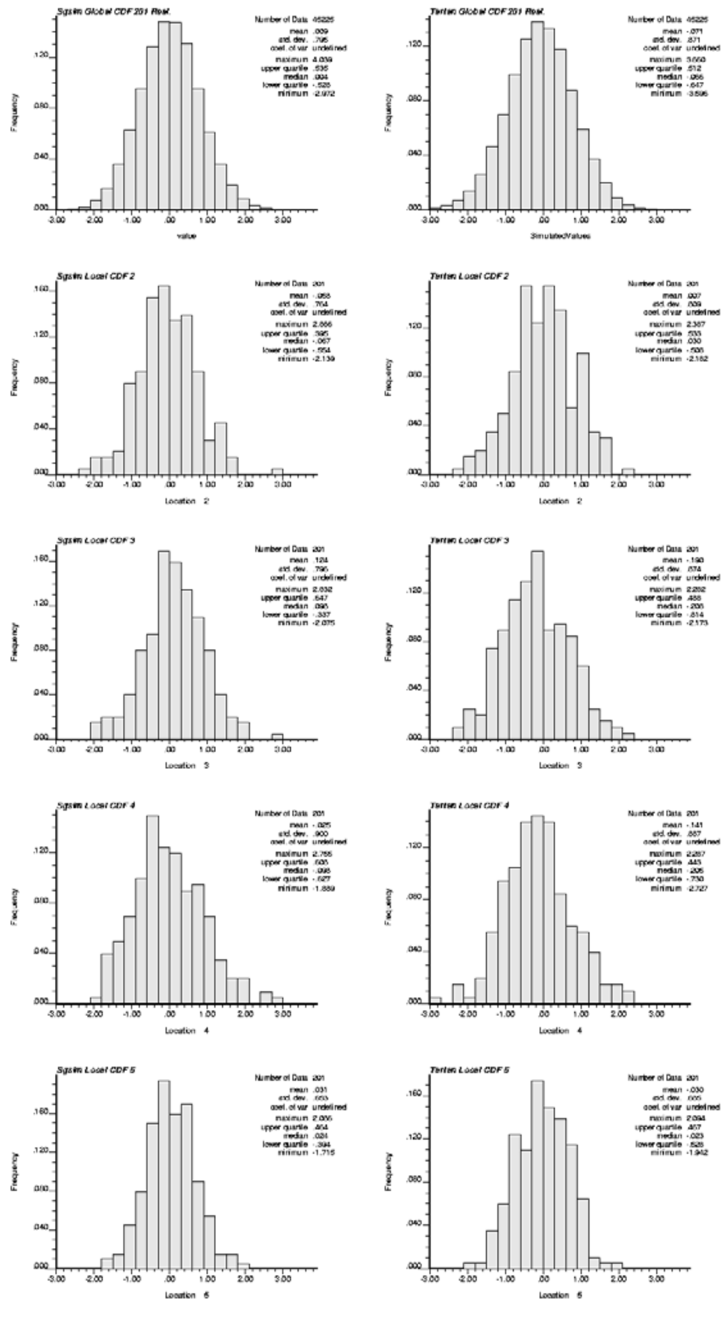

Figure 7: Some example local distributions of uncertainty compared between SGSIM block averaged (left) and TARTANSIM (right).

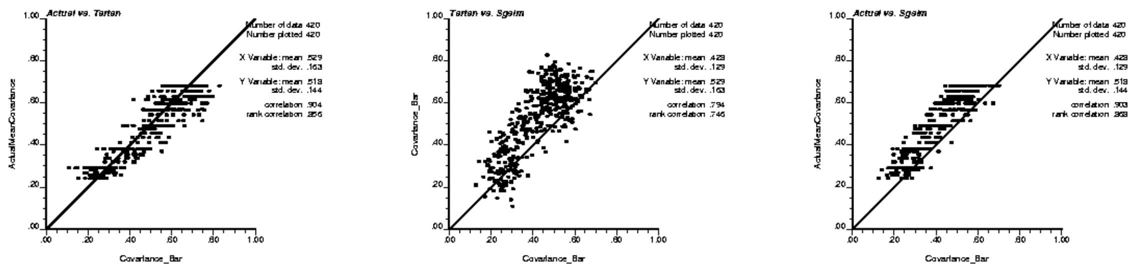

Figure 8: Scatter plots of the actual covariance (mean covariance calculated by 4x4 descritizations of the point variogram), and the empirical covariance calculated from 201 unconditional realizations of TARTANSIM and block averaged SGSIM.

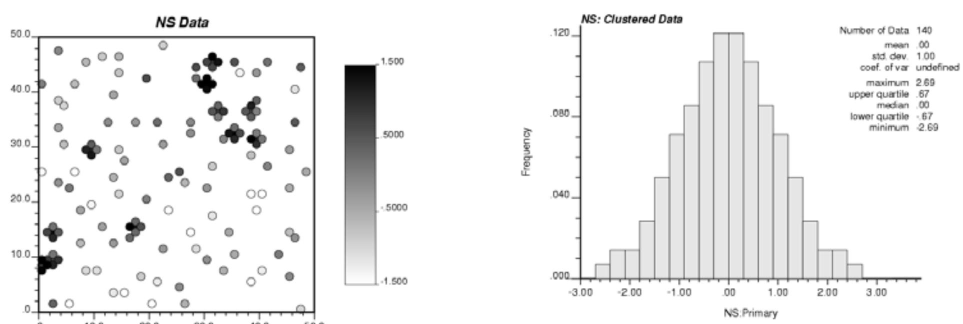

Figure 9: The red.dat data locations and Gaussian transformed distribution.

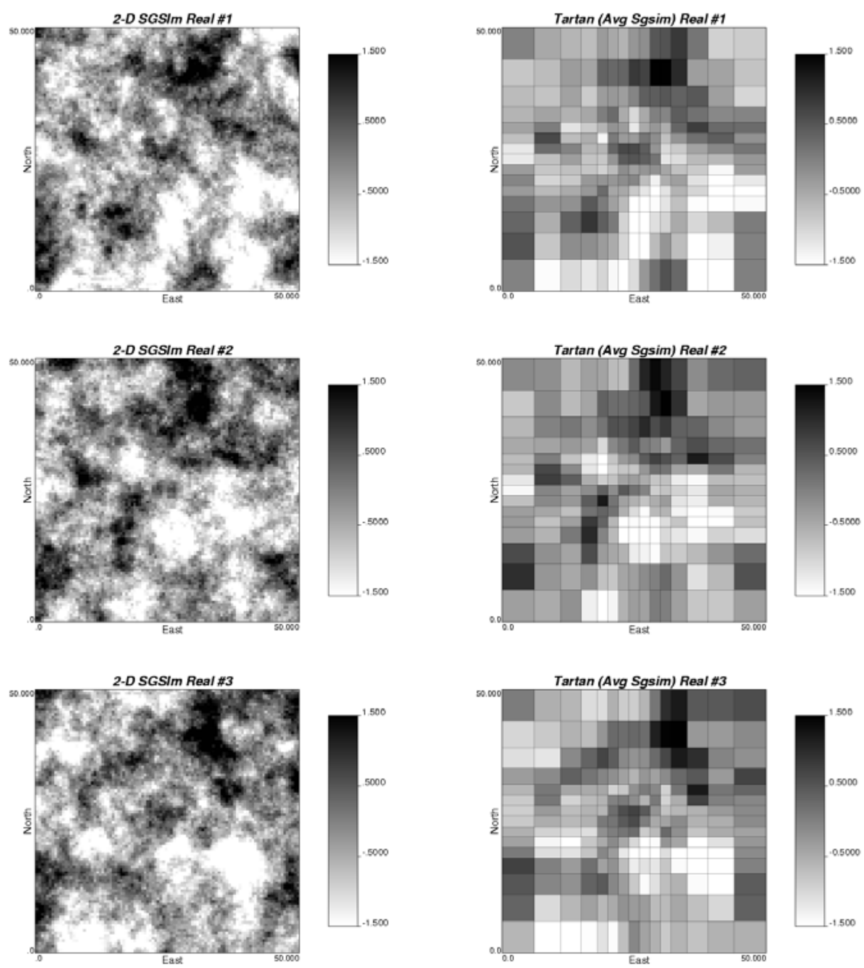

Figure 10: Example conditional SGSIM realizations and the associated block averaged maps.

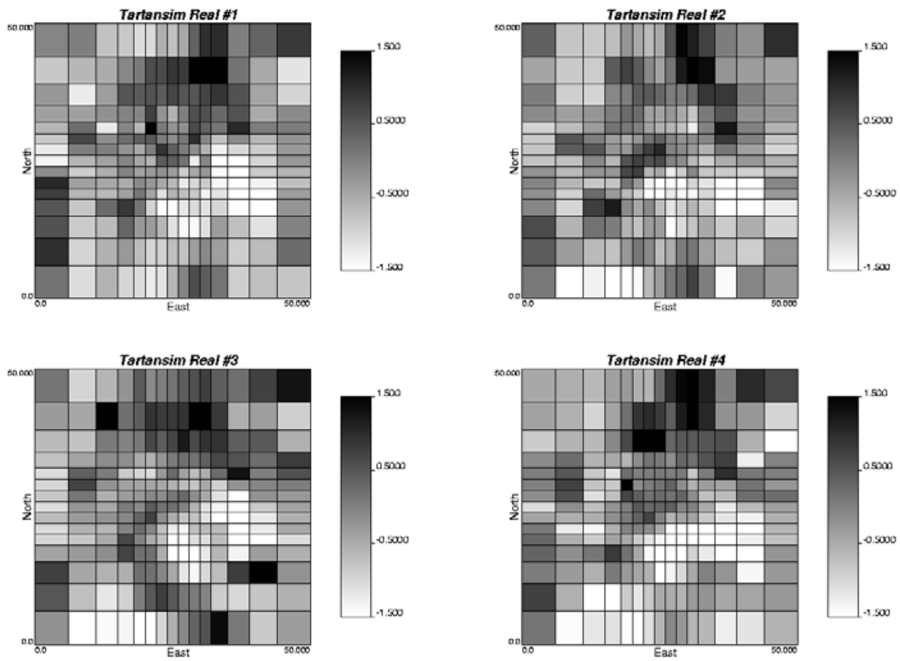

Figure 11: Example conditional TARTANSIM realizations.

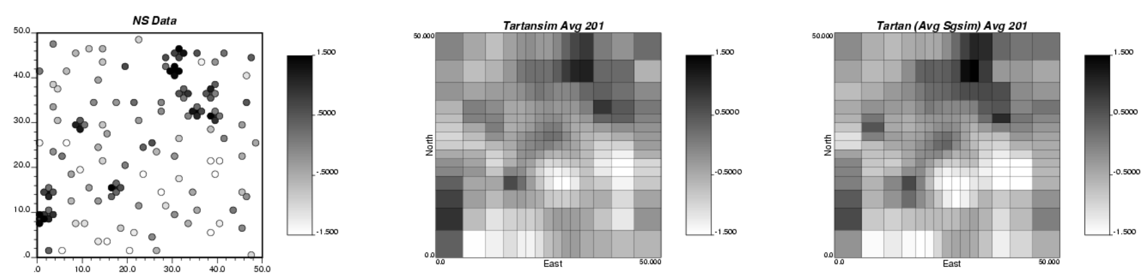

Figure 12: The conditioning data locations and the e-type estimates over 201 realizations for both TARTANSIM and block averaged SGSIM.

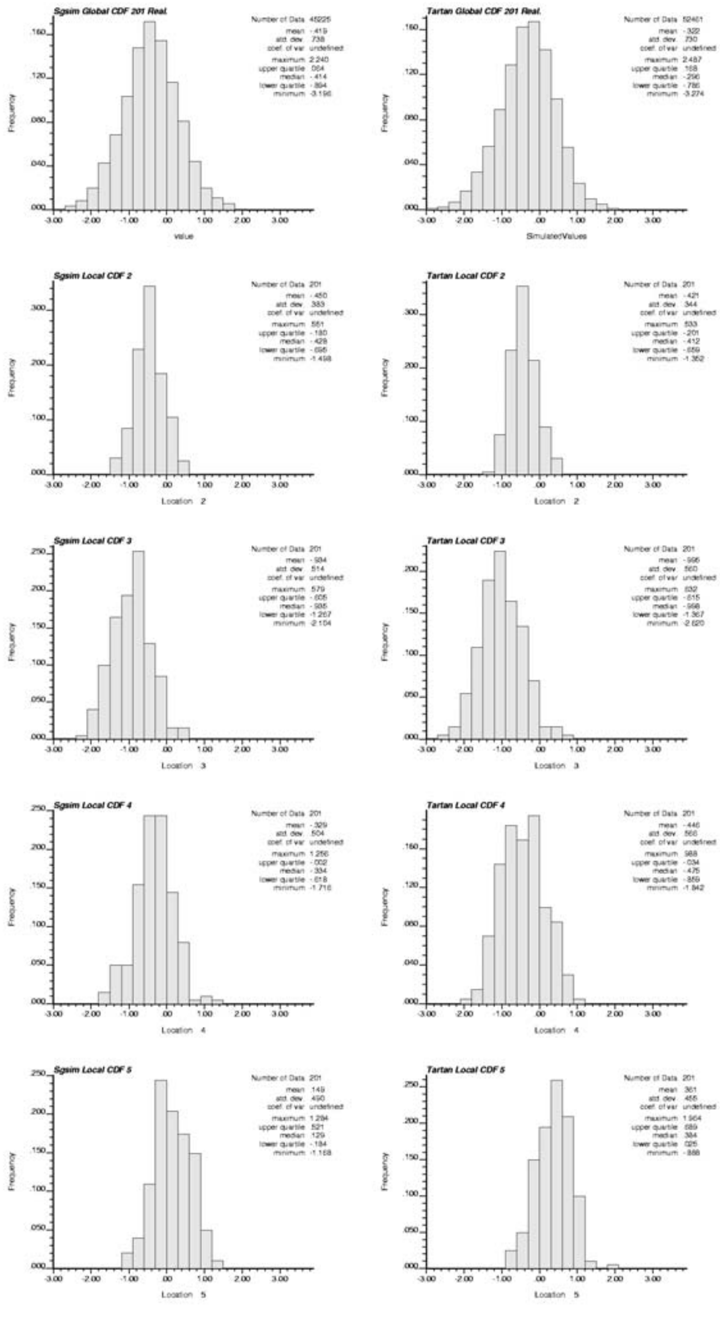

Figure 13: Some example local distributions of uncertainty compared between SGSIM block averaged (left) and TARTANSIM (right).

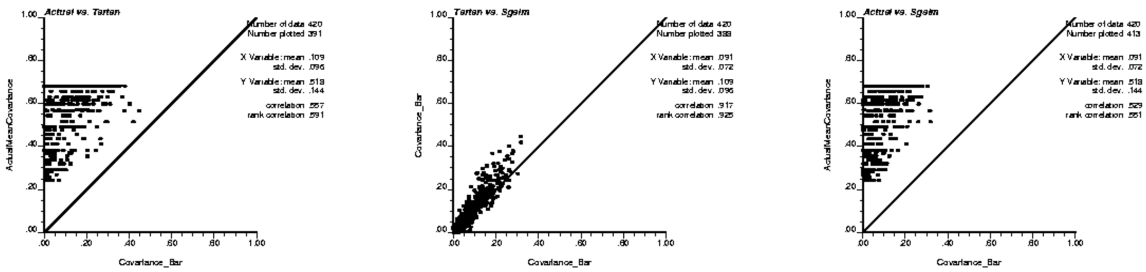

Figure 14: Scatter plots of the actual covariance (mean covariance calculated by 4x4 descritizations of the point variogram), and the empirical covariance calculated from 201 conditional realizations of TARTANSIM and block averaged SGSIM.

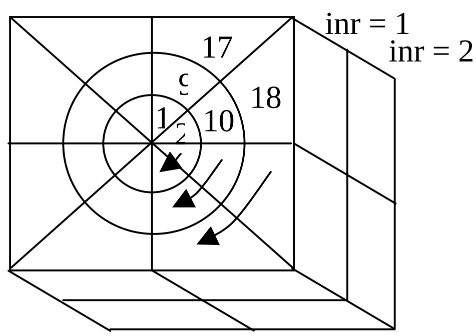

Figure 15: The nested radial grid format. The index proceeds first by sectors, then by tracks, and then by levels. The index begins at y axis and moves clockwise, the tracks begin on the inside and move outward and the levels begin at the bottom and move upwards.

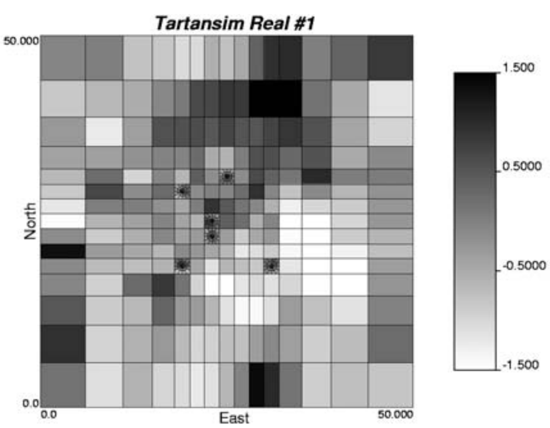

Figure 16: Example conditioned TARTANSIM realization with 6 nested radial grids.