A New Multiple Grid method in SIMPAT: a proposal

Hongmei Li, Burc G. Arpat and Jef Caers

Stanford Center for Reservoir Forecasting, Stanford University

Abstract

In order to model the multiple-scale, multiple-point correlations of complex geological structures, multiple-point geostatistics has been introduced as a promising field. The most recent research advance on this topic is the development of a new sequential simulation algorithm called Simulation with Patterns (SIMPAT), which redefines the geostatistical modeling as an image construction problem. SIMPAT is well suited to reproduce or mimic, in stochastic realizations, the complex multiple-scale patterns of a training image. To catch the multiple-scale nature of the training image patterns, SIMPAT relies on a multiple grid approach.

In this paper, we propose a new multiple-grid method to better reproduce such patterns. In the proposed multiple-grid method, multiple training images with different scales are used. The stochastic simulations are performed on multiple scales jointly instead of hierarchically. The main advantage of the proposed approach lies in its strong capability of handling scale relation and the possibility for new avenue in integrating multiple-scale data. The experimental result shows that it is feasible to implement the proposed algorithm in SIMPAT.
1. Introduction

In order to model multiple-scale, multiple-point correlations of complex geological continuity, multiple-point geostatistics has been developed as a promising field. The most recent research advance on this topic is the development of a new pattern-based geostatistical algorithm called SIMulation with PATterns (SIMPAT) (Arpat, 2005). The advantage of this new approach is that it focuses directly on the main purpose of reservoir modeling: reproduction of the multiple scale patterns in the final realization while anchoring them to the available subsurface data. To better handle the multiple-scale relations in the complex geological scenarios, SIMPAT introduces a new concept called dual template into the multiple-grid approach. In this paper, we propose an alternative multiple-grid approach that jointly simulates multiple-scale patterns by using multiple training images with different scales. The aim of the proposed method is to further improve the scale relations reproduction while conditioning different data at their right scales.

This paper starts with the motivation discussion, continues with the detailed description of the proposed multiple grid simulation algorithm, and ends with the experimental results using SIMPAT.

2. Motivation

This section briefly recalls the limitations of traditional multiple grid simulation approach, and reviews the multiple-grid method implemented in SIMPAT, then discusses the improvements that needed to better reproduce the multiple-scale relations of a complex training image.

2.1 Simulation using multiple grids

The traditional multiple grid approach (Tran, 1994) consists of simulating a number of increasing finer grids. The g-th (1 ≤ g ≤ n_g) grid is constituted by each 2^{g-1}-th node of the final (finest, g = 1) simulation grid. Accordingly, the coarse template (search
neighborhood) $T^g$ has the same structure configuration of nodes as the finest template $T$ ($g=1$) but with $2^{g-1}$ expansion space between the nodes.

The multiple-grid simulation starts from the coarsest grid $g=n_g$. The values simulated on the current grid are frozen and transferred to the next grid as conditioning data, then the next simulation is performed on grid $g-1$. This succession of various nested-grid simulations continues until the simulation of the finest grid $g=1$ is completed.

This multiple grid concept is proposed by Tran (1994) to reduce the CPU demand for the practical implementation of sequential simulation algorithm. According to the sequential simulation theory (Deutsch and Journel (1998)), each cCDFs should be conditioned to the original data and all previously simulated values. For large simulation grids, the cCDFs of the last simulated nodes have to be conditioned to a very large number of previously simulated values. Calculation of such conditional cCDFs becomes very CPU demanding. To overcome the CPU demand, one solution is dropping any conditioning data that does not fall within a given search template (neighborhood) centered at the current node to be simulated. Tran (1994) recognized that this data reduction may affect the reproduction of long range correlations, especially when the search template is small. Therefore, he proposed a multiple grid simulation solution which allows to capture large scale structure while considering a data search template with a reasonably small number of grid nodes thus save the CPU demand.

The traditional multiple grid approach works well under the assumption that the coarse sampling is representative of the complete, fine scale data event if applied successively (Arpat, 2005), i.e. the coarse data event is an approximation of the finest data event of the same size. This is true for simple patterns such as those observed in the high entropy realizations of a Multi-Gaussian law. However, when attempting to reproduce complex multiple scale geological patterns in a training image where different scales interact each other, this assumption may not hold. The same coarse pattern may correspond to several different fine scale patterns in the training image. Therefore, the traditional multiple grid approach cannot reproduce such scale relations of the complex, realistic training images.

The multiple grid approach implemented in SIMPAT is different from the traditional one in two ways. First, previously simulated values can be updated by the next grid
simulation. Once the current multiple grid simulation is completed, the values calculated on the current grid are transferred to the next grid. Instead of “freezing” the coarse grid values, the algorithm allows the subsequent multiple grid simulations to visit and revise these values. As a result, the values simulated on a coarse grid act as initial guesses for the finer grid simulation. Since coarse multiple grid realizations are assumed to contain the large scale information, this change results in the finer realization with better large scale connectivity. Second, to improve the level of approximation due to the use of coarse template, SIMPAT introduces a dual template method when performing multiple grid simulations. A dual template $\tilde{T}$ is defined as a template that it has the same bounding volume as primal template $T$ on grid $G$ but with all the finest nodes of the finest grid $G$ (Arpat, 2005, Figure 1-1). Therefore, the dual template is a fully and dense template. The dual pattern is defined as the pattern extracted at the same location $u$ in the training image using the dual template instead of the primal template (Arpat, 2005, Figure 1-1). In the pattern database, the primal and dual patterns extracted at the same location are stored as a pair. To use a dual template, first the most similar pattern is found based on the minimization of the distance between the primal data event and the primal pattern. Then the dual pattern is retrieved from the pattern database. Finally, instead of populating the nodes of primal data event, the algorithm populates the nodes of the dual data event. Once current multiple grid simulation is completed, the fully informed realization is used as the initial guess for the next multiple grid simulation. Then, finer grid simulations successively correct this guess. In other words, during simulation the large scale information is propagated to the finer grid using dual template. Therefore, the new multiple grid approach implemented in SIMPAT is more powerful for handling the scale interactions of complex geological scenarios.

Although the dual template method addresses the scale relation problem, it does not solve it. This is because the simulation is still hierarchical in nature: first simulate the coarsest scale, and then refine the finer scale realization. Furthermore, since the coarser patterns for the coarser grid are sub-sampled from the finest scale training image, they are not the true coarse scale patterns. These limitations call for the multiple grid simulations in parallel in terms of scale. If different scale patterns are simulated at one time, their relations could be reproduced more accurately since there is no approximation any more.
2.2 Simulation using multiple training images

The approach of using a different scale training image for each multiple grid has already been proposed by Strebelle (2000). In the case that a single complex training image containing the different scale relations is not available, one may use multiple independent training images: a large training image displaying the large scale patterns, and the smaller training images showing the finer scale patterns. For each multiple grid, a different training image is used. Because of the traditional multiple grid limitation stated in the previous section, this approach has the same shortcoming when attempting to reproduce very complex, multiple-scale patterns. The approach of Strebelle essentially
assumes that the finer scale patterns are a function of coarse ones. A nesting of scales is assumed which limits the type of geological heterogeneity being simulated.

Instead of simulating hierarchically, we propose to jointly simulate multiple-scale patterns using multiple training images with different scales at one time. This is different from the traditional approach where the large scale is simulated first and the smaller scales are simulated subsequently. The advantages of this joint simulation approach are:

1. Different scale training images or different upscaling techniques can be used to characterize several geological scales in one reservoir;
2. A feedback mechanism between fine scale and coarse scale is established. Different from the traditional multiple-grid simulation, where the coarse grid can affect the fine scale grid simulation, but fine scale cannot affect coarse scale;
3. Data conditioning at their appropriate scale becomes possible. Each type data has its own scale of information, hence affects the modeling of one or possibly multiple scales of heterogeneity of the underlying geology. The proposed idea creates new avenues for a truly multiple-scale data integrating method.

2.3 Notation in SIMPAT

In this section, some notation used in SIMPAT is recalled for explaining the proposed algorithm.

Define $t_i(u)$ as a value of the training image $t_i$ where $u \in G_i$, and $G_i$ is the regular Cartesian grid discretizing the training image. $t_i(T(u))$ indicates a specific multiple-point vector of $t_i(u)$ within a template $T$ centered at node $u$.

$\text{pat}_T^k$ is the particular $k$-th configuration of the previous vector of values $t_i(T(u))$ of the training image $t_i$, where $k = 1, \ldots, n_{\text{pat}_T}$, and $n_{\text{pat}_T}$ is the number of total available patterns in the pattern database $\text{patdb}_T$.

A data event $\text{dev}_T(u)$ is defined as the set of hard data and previously simulated values neighboring the visited location $u$ within the template $T$.

$$\text{dev}_T(u) = \{\text{dev}_T(u+h_1), \ldots, \text{dev}_T(u+h_a), \ldots, \text{dev}_T(u+h_{n_T})\}$$
where \(dev(u+h_a)=re(u+h_a)\) and \(re\) is the realization, \(dev_T(u) = re_T(u)\). \(h_a\) are the vectors defining the geometry of the \(n_T\) nodes of template \(T\) and \(a=1,\ldots,n_T\).

In SIMPAT, the similarity is measured through Manhattan distance function \(d(x, y)\) with \(d(x, y) \geq 0\) and \(d(x, y) = 0\) indicating complete similarity, i.e. \(x=y\). When using this notation, the data event is always written first, i.e. \(d(dev_T(u), pat_T^k)\) meaning the distance between data event \(dev_T(u)\) and pattern \(pat_T^k\).

3 Proposed method-simulation in parallel

A new multiple grid approach is proposed based on the multiple training image concept. The proposed approach simulates the multiple grids in parallel. The algorithm for unconditional simulation is described in detail.

3.1 Proposed algorithm for unconditional simulation

Consider that we have a different training image for each multiple grid to be simulated, the training image \(t_i\) now is a vector \(t_i\_vec\) (Figure 3-1), each component corresponding to a different scale training image \(t_i_g\) (\(g=1,\ldots,n_g\)). Since the ‘training image’ is a vector, the template \(T\) used for scanning the training image is a vector \(T\_vec\) as well (Figure 3-1). Each template component \(T_g\) is used to scan the training image component \(t_i_g\). The templates have following features:

1. They are fully informed, there is no expansion, while in traditional multiple grid approach, the template nodes for coarse grid \(G^g\) are “expanded” with \(2^{g-1}\) spacing;
2. They may have different dimensions for each multiple grid;
3. The template components share the same central location;

Giving the training image vector and the template vector, multiple scale patterns are extracted (Figure 3-2). For a location \(u\) (red point in Figure 3-2), the center of template component (black square in Figure 3-2) is the same for all training image components. The patterns extracted by the template vector are stored as a vector as well in the pattern
database. Since each time the extraction is performed on multiple training images, each pattern vector $\text{pat}_k^T\text{vec}$ contains multiple patterns with different scales. These patterns do not contain blank nodes. The preprocessing is exactly the same as SIMPAT, except that the training image vector $\text{ti}_\text{vec}$ consists of multiple training images instead of one by one preprocessing.

![Coarse Fine](image)

**Figure 3-1** Multiple training images (top) with different size template (bottom)

![Pattern extraction](image)

**Figure 3-2** Pattern extraction is performed jointly. Each template component scans the same location of its corresponding training images. The patterns are stored as one vector.

Consequently, the simulation is conducted jointly. For all multiple grid, at every node $u \in G_{re}$ of realization $\text{re}$, the data event for all grids $G_{re}$ are extracted at one time, forming a data event vector $\text{dev}_\text{vec}(u)$. Then the data event vector is compared to all available pattern vectors $\text{pat}_\text{vec}^T$ in the pattern database $\text{patdb}_T$ using a distance function. The comparison is performed between the data event components and the pattern components at the same scale. In other words, the similarity calculation is
performed on all scales jointly. The distance between a data event vector and a pattern vector is the sum over all absolute difference of all scale components:

\[ d\left\{ \text{dev}_{\text{T}}(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\alpha=0}^{n_{\text{T}}} d_{\alpha} \left\{ \text{dev}_{\text{T}}^\alpha(u), \text{pat}_{\text{T}}^k \right\} \]

For example using the Manhattan distance,

\[ d_{\alpha} \left\{ \text{dev}_{\text{T}}^\alpha(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\beta=0}^{n_{\text{g}}} \left| \text{dev}_{\text{T}}^\alpha(u + \mathbf{h}_\beta) - \text{pat}_{\text{T}}^k(\mathbf{h}_\beta) \right| \]

where \(d\left\{\ldots\right\}\) is the Manhattan distance, \(n_{\text{g}}\) is the number of multiple grid and \(n_{\text{T}}\) is the number of node in template \(T^g\).

For example, for a four multiple-scale training image case, a pattern vector contains 4 scale patterns (\(n_{\text{g}}=4\)), and a data event vector has 4 scale data events (Figure 3-3). The Manhattan distance is the sum of the absolute difference between each data event and its corresponding scale pattern:

\[ d = d_1 + d_2 + d_3 + d_4 \]

Where

\[ d_1 = d_{1} \left\{ \text{dev}_{\text{T}}^1(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\beta=0}^{n_{\text{g}}} \left| \text{dev}_{\text{T}}^1(u + \mathbf{h}_\beta) - \text{pat}_{\text{T}}^k(\mathbf{h}_\beta) \right| \]

\[ d_2 = d_{2} \left\{ \text{dev}_{\text{T}}^2(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\beta=0}^{n_{\text{g}}} \left| \text{dev}_{\text{T}}^2(u + \mathbf{h}_\beta) - \text{pat}_{\text{T}}^k(\mathbf{h}_\beta) \right| \]

\[ d_3 = d_{3} \left\{ \text{dev}_{\text{T}}^3(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\beta=0}^{n_{\text{g}}} \left| \text{dev}_{\text{T}}^3(u + \mathbf{h}_\beta) - \text{pat}_{\text{T}}^k(\mathbf{h}_\beta) \right| \]

\[ d_4 = d_{4} \left\{ \text{dev}_{\text{T}}^4(u), \text{pat}_{\text{T}}^k \right\} = \sum_{\beta=0}^{n_{\text{g}}} \left| \text{dev}_{\text{T}}^4(u + \mathbf{h}_\beta) - \text{pat}_{\text{T}}^k(\mathbf{h}_\beta) \right| \]

Figure 3-3 The distance between a data event vector and a pattern vector is the sum of their four components distance
Following is the proposed algorithm describing previous process:
Step 1 Generate multiple scale training images or a training image vector
\[ \text{ti}_\text{vec}=\{\text{ti}_1, \text{ti}_2, \ldots, \text{ti}_n \}; \]
Step 2 Create multiple scale templates or a template vector \[ \text{T}=\{T^1, T^2, \ldots, T^n \}; \]
Step 3 Scan the training image vector \[ \text{ti}_\text{vec} \] using template vector \[ \text{T}_\text{vec} \] for the grid vector \[ \text{G}_\text{vec}_\text{ti} \] to obtain all existing patterns \[ \text{pat}_\text{vec}^k; \text{T}^k, k=1, \ldots, n_{\text{pat}} \] that occur over the \[ \text{ti} \]; these patterns form the pattern database \[ \text{patdb}_{\text{T}} \];
Step 4 Define a random path on grid \[ \text{G}_\text{re} \] of the realization \[ \text{re} \] to visit each node \[ \text{u} \] only once;
Step 5 At each node \[ \text{u} \], extract the data event \[ \text{dev}_\text{vec}_\text{T}(\text{u}) \] from the realization \[ \text{re} \] and find the \[ \text{pat}_\text{vec}_\text{T}^* \] that minimizes the \[ d(\text{dev}_\text{vec}_\text{T}(\text{u}), \text{pat}_\text{vec}_\text{T}^k) \] for \( k=1, \ldots, n_{\text{pat}} \) in the pattern database \[ \text{patdb}_{\text{T}} \]. In other word, \[ \text{pat}_\text{vec}_\text{T}^* \] is the most similar pattern vector to \[ \text{dev}_\text{vec}_\text{T}(\text{u}) \];
Step 6 Once the most similar pattern \[ \text{pat}_\text{vec}_\text{T}^* \] is found, past it to \[ \text{dev}_\text{vec}_\text{T}(\text{u}) \];
Step 7 Move to the next node of the random path and repeat the above steps until all the grid nodes along the random path are visited.

In the above algorithm, the concepts of training image, template, pattern, data event are the same as the ones defined in SIMPAT. The only difference is that here multiple training images are used instead of single training image used in SIMPAT. As a result, all the parameters such as template, data event and patterns are present as vectors. The simulation is performed in parallel in terms of scales, it is no longer hierarchical.

The big challenge here is the search for the most similar pattern \[ \text{pat}_\text{vec}_\text{T}^* \]. Since the multiple training images are used during multiple grid simulation at one time, the similarity search involves comparing of huge number of data event vector and pattern vectors in pattern database. In this case how to efficiently find the most similar pattern vector is a challenge. SIMPAT does not use any cluster or classification methods, the similarity search could be very slow when the pattern database contains a large number of patterns. To solve this problem, dimension reduction techniques such as cluster or classification (Hart and Stork, 2001) and optimum search strategy such as different search trees (Salembier and Garrido, 2000) have to be used for the multidimensional...
database. This paper does not cover the specific dimension reduction technique used for the proposed approach, instead, it will be part of the future research.

4. Experiments

A few initial experiments of this idea have already been conducted and show some promising results. Since we currently are not able to fully implement the algorithm described above, these experiments are carried out using a modification of SIMPAT’s current multiple-grid approach. To demonstrate the experimental procedure, an unconditional simulation is performed using a 180x120 training image given in Figure 4-1[a] and four multiple-grids.

(1) Generate 4 training images for 4 multiple-grids simulation

For the training image of Figure 4-1[a], three other coarse scale training images are obtained (Figure 4-1[b]-[d]) by using 5x5, 9x9 and 15x15 moving average respectively. These four training images (Figure 4-1, [a]-[d]) are named $t_i$, $t_i$, $t_i$, $t_i$ in the order of coarsening scale. They have the same dimension, i.e. 180x120.

(2) Create 4 templates for four multiple-grids

A 7x7 template is chosen as base template $T^1$. This template is expanded by $2^3$ to get a 49x49 coarse template $T^4$, by $2^2$ to obtain a 25x25 template $T^3$, and by $2^1$ to get a 13x13 template $T^2$;

(3) Simulation starts from the 4th (the coarsest) multiple-grid.

First, the patterns are extracted from $t_i$ and $t_i$ jointly using template $T^4$. As a result, the pattern database $\text{patdb}_T$ stores pairs of patterns per index $k$ ($k = 1, \ldots, n_{\text{pat}}$), i.e.

\[
\{ \text{pat}^k_{T^4, u_i}, \text{pat}^k_{T^4, u_i} \}. 
\]

Next, the SIMPAT single-grid unconditional simulation algorithm (Algorithm 3.2, Arpat, 2005) is modified to obtain an approximation of the proposed idea. The first modification here is that at every node $u$ of grid $G_{re}$, there are two data events: coarse data event $\text{dev}^4_{T^4}(u)$ from grid $g=4$ and finer data event $\text{dev}^3_{T^4}(u)$ from grid $g=3$. 

11
Consequently, the second modification is that the most similar pattern pair $(\mathbf{pat}_{T_4}^e, \mathbf{pat}_{T_4}^e)$ is found which minimizes:

$$d = d_4 \left\langle \mathbf{dev}_{T_4}^e(u), \mathbf{pat}_{T_4}^e \right\rangle + d_3 \left\langle \mathbf{dev}_{T_3}^e(u), \mathbf{pat}_{T_3}^e \right\rangle$$

Having these modifications, the single-grid unconditional simulation algorithm is performed on the 4th multiple-grid which has dimension of 180x120. Once the simulation is completed, there are two outcomes: a coarse grid realization (Figure 4-2 [c]) reproducing the patterns of $\mathbf{ti}_4$ and a finer grid realization (Figure 4-2[d]) reproducing the patterns of $\mathbf{ti}_3$. The finer scale realization will be used as the initial guess of the next multiple-grid simulation.

(4) Next multiple simulation is conducted on the 3rd 180x120 multiple-grid. The pattern extraction and the simulation algorithms are the same as the ones described in step 3 except that (1) this time the template $T_3$ is used for pattern extraction and following simulation; (2) the patterns are extracted from $\mathbf{ti}_2$ and $\mathbf{ti}_3$ jointly; (3) the simulation starts on a fully informed grid instead of a fully unknown grid, because the finer grid realization in step 3 is used as the initial guess of current grid. Figure 4-3[c] and [d] are the resulting realizations.

(5) Similarly, the 2nd multiple-grid uses template $T_2$ and patterns are extracted from training image $\mathbf{ti}_2$ and $\mathbf{ti}_1$. The modified simulation algorithm works on the initialized grid with dimension of 180x120. The simulated results are shown in Figure 4-4[c] and [d].

(6) Accordingly, the next and last (1st) 180x120 multiple-grid takes the finer grid result of Figure 4-4[d] as the initial guess. The base template $T_1$ and the training image $\mathbf{ti}_1$ are used for this grid. The simulation is performed in the same way as previous one. The outcome of this step (Figure 4-5[b]) represents the output realization. As we can see, this final realization reproduces the patterns shown in the original training image (Figure 4-1[a]) very well.

The main difference between the modified SIMPAT multiple-grid approach and the proposed one is that the former uses two training images at a time instead of all the multiple training images at one time. In SIMPAT, the “training image bands” concept (Arpat², 2005) allows to set multiple training images as multiple bands and then to simulate them jointly. The reason for simulation using two training images at one time is
because of the CPU demand consideration. SIMPAT uses direct search to find the most similar pattern. When simulations are performed for multiple training images in parallel, the pattern vectors have much large dimension than a single pattern. Therefore, the similarity search will be extremely slow at this stage. In our experiments, this CPU demand is less of problem because only two simulations are conducted at one time. As shown in the example above, even using two training images at one time, the simulated results successfully capture the patterns of the training image. These promising results indicate that the proposed algorithm has potential to improve the multiple-scale patterns reproduction quality when it applies to more complex geological scenarios.

Comparing the modified algorithm with SIMPAT’s original one, the first difference is the scale of the initial guess. As we stated before, SIMPAT uses dual template simulation to obtain an initial guess for the finest realization based on the coarse information provided in the coarsest grid, then the next finer grid simulations are performed to “correct” this guess. In other words, the initial guesses for different multiple-grids have the same finest scale. While in the modified algorithm, the initial guesses for different multiple-grids have different scales due to using different training images. The second difference is the quality of the initial guess. The modified algorithm generates the initial guesses by using the similarity search, while in SIMPAT’s algorithm, the initial guesses are obtained without directly involving with similarity calculations. As a result, the initial guess obtained using the modified algorithm is much closer to its final realization compared with the SIMPAT case. Hence it is a better initialization. Figure 4-6 shows one of the realizations generated by SIMPAT original algorithm using the training image of Figure 4-1[a] and 4 multiple-grids. The simulation uses the same base template (7x7) and the same distance function (Manhattan distance) as the ones used in the above experiment. Comparing Figure 4-6 with Figure 4-5[b], we can see that the former does not reproduce the patterns of the training image. According to Arpat (2005), this is because the categories (fault and layer) in the training image have low proportion, thus the Manhattan distance fails to pick the “right” pattern.
Figure 4-1 Original training image containing fault-layer-background categories and different scale training images obtained by moving average.

Figure 4-2 Training images used and resulting realizations of 4th multiple-grid
Figure 4-3 Training images used and resulting realizations of 3rd multiple-grid

Figure 4-4 Training images used and resulting realizations of 2nd multiple-grid
5. Conclusions and future work

(1) A new approach for simulating complex patterns using multiple training images at multiple scales is proposed. The approach aims to improve the approximations made in the use of coarse template;

(2) Experimental study shows the multiple training images approach is feasible in SIMPAT, and the final results successfully reproduce the patterns of the training image;

(3) Future research will focus on the similarity search strategy. The objective is to find an adequate clustering method. Then an optimal search strategy will be adopted in SIMPAT based on this method;

(4) Data conditioning will be studied as it is an important part of the reservoir modeling;

(5) A 3D case study with real data.
6. Reference


