Flow in a reservoir is mostly controlled by the connectivity of extreme permeabilities which are generally associated with marked geological patterns. Thus, accurate characterization of such patterns is required for successful flow performance and prediction studies.

Multiple-point geostatistics (MPS), a relatively new branch of geostatistics, manages to achieve the desired pattern reproduction while successfully constraining to local subsurface data. MPS uses the concept of training image to quantify geological heterogeneities. In MPS, a training image is defined as a conceptual depiction of the type of heterogeneities that the geologist believes to exist in the reservoir, i.e. it reflects a prior geological concept. MPS utilizes the training image to export multiple-point statistics and from thereon follows the familiar extended sequential simulation framework.

Within this context, a new pattern-based geostatistical approach is proposed that redefines reservoir characterization as an image construction problem. The approach utilizes the training image concept of MPS but instead of exporting multiple-point statistics, it infers multiple-scale geological patterns that occur within the training image and uses these patterns as the building blocks for image (reservoir) construction. Each pattern inferred corresponds to a multiple-pixel configuration that identifies a piece of a geological shape that exists in the conceptual geological model. Such patterns might be at different scales and might interact with each other. Once the patterns are inferred from the training image, the method applies sequential simulation to reproduce these patterns in the realization via honoring a similarity (distance) function. Multiple-scale interactions of patterns are handled through a new multiple-grid approach. The method achieves better pattern reproduction and large scale continuity than existing pixel-based reservoir characterization techniques and works equally well with both continuous and categorical variables (such as permeability and facies) while constraining to a variety of local subsurface data such as well logs and 3D seismic.

Introduction

Sequential simulation is one of the most widely used stochastic imaging techniques within the Earth Sciences. The sequential simulation idea found its first use in the traditional variogram-based geostatistics algorithms such as sequential Gaussian simulation (SGSIM) and sequential indicator simulation (SISIM). In the early nineties, multiple-point geostatistics (MPS) introduced the training image concept proposing to replace the variogram with the training image within an extended sequential simulation framework (Guardiano and Srivastava, 1993).

---

G. Burc ARPAT and Jef CAERS

Dept. of Petroleum Engineering, Stanford Univ., Stanford CA, 94305-2220, USA
MPS manages to achieve the desired shape reproduction by replacing the variogram with a training image. In MPS, a training image is defined as a conceptual depiction of the type of heterogeneities that geologist believes to exist in the field of study. Training images used by MPS need not be conditional, i.e. carry any locally accurate information as provided by location-specific data. They only reflect a prior structural concept. In Guardiano and Srivastava (1993), the authors propose a sequential simulation algorithm where one calculates the conditional probabilities (ccdfs) through sampling the available neighborhood information (called a “data event”) from the training image. Simply put, the algorithm extracts the neighborhood information (data event) using a template (window) and looks for replicates of this data event in the training image. The ccdf is calculated by mere counting of the frequency of the event occurrence in the center node of these replicates. The original approach of Guardiano and Srivastava was CPU demanding due to the rescanning of the training image for every uninformed node on the simulation grid. Later, Strebelle (2000) proposed to use a special data structure (called a “search tree”) to store training image events for fast lookup. The algorithm of Strebelle is called SNESIM and presents a landmark development in multiple-point geostatistics, being the first truly practical algorithm (See Caers et al. (2003) for a large-scale example).

The main limitation of MPS lies in its requirement of “stationarity” of the training image, which is inherent to any stochastic simulation tool based on the Probability Theory. In MPS, the conditional probabilities of the data events are calculated by scanning the training image for “repetitions” or replicates of the same event. This requires a form of stationarity (translation invariance of multiple-point statistics) of the training image that would arguably limit the use of MPS. Some solutions have been provided to address this problem. Yet, they fail to address all types of non-stationarities that may exist in a field of study (Caers and Arpat, 2004). There is no evident solution for stationarity within the framework of probability since every single method that relies on calculating or modeling probabilities will have to rely on stationarity.

On large simulation grids, another practical problem occurs when deriving the ccdf due to the increasing amount of conditioning information as more nodes are visited. The ccdf at the last simulated node would have to be conditional to all previously simulated values as per the theory of sequential simulation (Deutsch and Journel, 1998). All sequential simulation methods, including SGSIM, SISIM and MPS suffer from this problem. To solve the problem, practical implementations of sequential simulation algorithms rely on the following two solutions: First, any conditioning data that does not fall with a given search neighborhood is discarded. Reducing the number of data makes inference of the ccdfs easier, but as recognized by Tran (1994), may degrade the reproduction of long range correlations. Therefore, in addition to dropping the furthest data values, Tran proposes to use a so-called multiple-grid method. On a Cartesian grid, the idea is to use a set of cascading multiple-grids and templates (neighborhoods) instead of a single grid and one large dense template. Algorithms using the multiple-grid approach proceed first by performing sequential simulation on the coarsest grid. The values simulated on the coarse grid are frozen and passed to subsequent finer grids as conditioning data.

Some of the shortcomings of the above multiple-grid method have been recognized when attempting to reproduce complex geological continuities. In the multiple-grid approach, due to its hierarchy from coarse to fine, the fine grid simulated values are conditioned by the coarse grid ones, but not vice-versa. In attempting to reproduce interdependent, continuous geological heterogeneity, such a multiple-grid approach may not be desired.

A New, Pattern-based Sequential Simulation Method

An alternative and complementary approach to the sampling strategy of Guardiano and Srivastava and Strebelle is to redefine the problem as an image processing problem. In image
processing, one generally tackles complex images by finding common patterns in the image and working on these patterns (Palmer, 1999). A similar approach can be devised for geostatistical modeling where one finds all the patterns of a training image given a template. These patterns correspond to multiple-pixel configurations that identify meaningful pieces of geological shapes known to exist in field of study. Such patterns exist at different geological scales and patterns at various scales interact with each other. The idea is to generate realizations that reproduce these multiple-scale patterns on the simulation grid. This idea is implemented and named SIMPAT (SIMulation with PATterns) and is elaborated in the following paragraphs, without considering the application of the multiple-grid approach, which is explained later.

The algorithm starts by scanning the training image using a template (window) to acquire all patterns. A filter can be applied to discard the patterns with undesirable characteristics. Remaining patterns are stored in a pattern database and such patterns are denoted by $\text{pat}_T^k$ where $T$ denotes the template used and $k$ is the pattern index and $k = 0, \ldots, n_{\text{pat}} - 1$ (total number of patterns in the database).

Then, the algorithm carries on with the simulation, following the extended sequential simulation framework. During simulation, the neighboring hard data and previously simulated nodes of a visited grid node $u = (x,y,z)$ is extracted (denoted as $\text{dev}_T(u)$; the data event) and compared to all available patterns in the pattern database using a predefined similarity criterion. The aim is to find the ‘most similar’ pattern to the data event, denoted by $\text{pat}_T^*$. Once the ‘most similar’ pattern is found, the data event $\text{dev}_T(u)$ is replaced by $\text{pat}_T^*$, i.e. the values of $\text{pat}_T^*$ is pasted on to the simulation grid at the current node $u$.

The similarity between a data event and a pattern is calculated using a pixel-based distance function. If data event $\text{dev}_T(u)$ is defined as,

$$\text{dev}_T(u) = \{i(u + h_0), i(u + h_1), \ldots, i(u + h_{n_T})\}$$

(1)

where $u$ denotes the current node, $i$ is the simulation grid and $h_\alpha$ are the vectors defining the geometry of the template $T$ with $\alpha = 0, \ldots, n_T$ (number of nodes in the template); and, a pattern $\text{pat}_T^k$ is defined as,

$$\text{pat}_T^k = \{\text{pat}_T^k(h_0), \text{pat}_T^k(u + h_1), \ldots, \text{pat}_T^k(u + h_{n_T})\}$$

(2)

where $\text{pat}_T^k$ denotes a node of the location independent pattern definition; then, a distance between the data event $\text{dev}_T(u)$ and the pattern $\text{pat}_T^k$ can be defined as,

$$d(\text{dev}_T(u), \text{pat}_T^k) = \sum_{\alpha=0}^{n_T-1} |\text{dev}_T(u + h_\alpha) - \text{pat}_T^k(h_\alpha)|$$

(3)

where $d<>$ denotes the distance function, also known as the Manhattan or the “city block” distance (Duda et al., 2001). For other distance functions that can be used with the SIMPAT algorithm, see Arpat (2004).

The above outlined algorithm can be divided into two main parts: (1) preprocessing of the training image and (2) simulation on the simulation grid (realization).

Preprocessing of the training image:

P-1. Scan the training image using the template $T$ to obtain all existing patterns $\text{pat}_T^k$ that occur over the training image.

P-2. Reduce the number of patterns to $n_{\text{pat}}$ by applying filters to construct the pattern database. Typically, only unique patterns are taken.
Simulation on the simulation grid:

S-1. Define a random path on the simulation grid to visit each node \( u \) only once.

S-2. At each node \( u \), retain the data event \( \text{dev}_T(u) \) and find the \( \text{pat}_T^* \) that minimizes \( d(\text{dev}_T(u), \text{pat}_T^k) \) for \( k = 0, \ldots, n_{\text{pat}} - 1 \), i.e. \( \text{pat}_T^* \) is the ‘most similar’ pattern to \( \text{dev}_T(u) \).

S-3. Once the most similar pattern \( \text{pat}_T^* \) is found, assign \( \text{pat}_T^* \) to \( \text{dev}_T(u) \), i.e. for all the \( n_T \) nodes \( u + h_\alpha \) within the template \( T \), \( \text{dev}_T(u + h_\alpha) = \text{pat}_T^*(h_\alpha) \).

S-4. Move to the next node of the random path and repeat the above steps until all the grid nodes along the random path are exhausted.

As described above, the method applies only to a single grid. Application of the new multiple-grid approach is discussed below.

**A New Multiple-grid Approach**

On a Cartesian grid, the multiple-grid view of a grid is defined by a set of cascading coarse grids and templates \( T^g \) instead of a single fine grid and one large dense template where \( g = 0, \ldots, n_g - 1 \) and \( n_g \) is the total number of multiple-grids. The \( g \)-th \( (0 \leq g \leq n_g - 1) \) coarse grid is constituted by each \( 2^g \)-th node of the final grid \( (g = 0) \). If \( T \) is a template defined by vectors \( h_\alpha \), then the template used for a coarser grid \( T^g \) is defined by \( h_\alpha^g = 2^g \times h_\alpha \) and has the same configuration of \( n_T \) nodes as \( T \) but with spacing \( 2^g \) times larger.

The multiple-grid simulation of a realization is achieved by successively applying the single-grid algorithm explained above to the multiple-grids starting from the coarsest grid. After each multiple-grid simulation, the values calculated on the current grid are transferred to the one finer grid and \( g \) is set to \( g - 1 \). This succession of multiple-grid simulations continues until \( g = 0 \). It should be noted that, different from the classical multiple-grid approach (Tran, 1994), SIMPAT does not ‘freeze’ the coarse grid values when they are transferred to a finer grid, i.e. such values are still allowed to be updated and visited by the algorithm in the subsequent multiple-grid simulations. On a multiple-grid, the previously calculated coarser grid values contribute to the distance calculation of Step S-2, i.e. if on node \( u \) a previous coarse grid value exists, this value is taken into account when minimizing the distance.

The above multiple-grid approach allows the values determined on the coarser grids to be modified by the finer grids given the newly available information, i.e. top to bottom interaction. SIMPAT also utilizes a feedback mechanism that allows bottom to top interaction. The reader is referred to Arpat (2004) for details of this approach.

**Primary Data Conditioning**

In SIMPAT, conditioning to primary data (such as well data) is performed in Step S-2 of the algorithm, during the search for the most similar pattern. If conditioning data exists on any \( \text{dev}_T(u + h_\alpha) \), the algorithm first checks whether \( \text{pat}_T^k(h_\alpha) \) is equal to this data. If the pattern \( \text{pat}_T^k \) does not fulfill this condition (i.e. there is a mismatch), it is skipped and the algorithm searches for the next most similar pattern until a match is found. If none of the available patterns fulfill the condition, the algorithm selects a pattern such that only the nodes of the data event that has conditioning information are considered during the distance calculations and other nodes are ignored. Note that, such a case generally occurs when the training image is not representative of the conditioning data. If several patterns fulfill this condition, then a second minimization is performed on the non-conditioning nodes of the data event using only these patterns.
The above conditioning methodology can be summarized as follows:

1. At node \( u \), retrieve the data event \( \text{dev}_T(u) \). Divide the data event into two parts: \( \text{dev}_{T,a} \) and \( \text{dev}_{T,b} \) such that \( \text{dev}_{T,a} \) contains only the conditioning nodes of \( \text{dev}_T(u) \) and \( \text{dev}_{T,b} \) contains only the non-conditioning and uninformed nodes of \( \text{dev}_T(u) \).

2. Minimize the distance between \( \text{dev}_{T,a} \) and \( \text{pat}_T^k \) to find the most similar pattern \( \text{pat}_T^* \) to the conditioning data event. If only one such pattern exists, take \( \text{pat}_T^* \) as the final most similar pattern and proceed with Step S-3 of the SIMPAT algorithm.

3. If several patterns minimize the above distance, perform another minimization, namely the distance between \( \text{dev}_{T,b} \) and \( \text{pat}_T^k \), only within these conditioned patterns to find the final most similar \( \text{pat}_T^* \).

**Secondary Data Conditioning**

In Earth Sciences, secondary data typically refers to data obtained from an indirect measurement such as seismic surveys. Thus, secondary data is nothing but a ‘filtered’ view of the original field of study (reservoir), where the filtering is performed by some forward model \( F \). Generally, the forward model \( F \) is not fully known and is approximated by a known model \( F^* \).

Within this context, in SIMPAT, conditioning to secondary data calls for additional concepts to be introduced the basic algorithm. First, a secondary training image is required. This training image can be obtained by applying the approximate forward model \( F^* \) to the training image. The rationale for this requirement is as follows: The patterns of the secondary data (For example, seismic impedance) are only related to the geological (primary) patterns of the realization through the above mentioned filter model. Typically, this model operates on the entire grid (For example, seismic filters consider the overburden and other concepts such as the Fresnel zone). Thus, modeling one to one relations of the primary and secondary patterns independent of the grid they exist in is not possible.

Once the secondary training image is obtained, SIMPAT explicitly relates the patterns in the primary training image and the secondary training image in the pattern database. In other words, Step P-1 of the algorithm is modified such that, for every \( \text{pat}_T^k \) of the primary training image, a corresponding secondary pattern is extracted from the secondary training image from the same location \( u \). In essence, patterns now exist in pairs in the pattern database. Such explicit relating of patterns for the specific grids allows the user of the algorithm to indirectly introduce the physics of the filtering to the simulation without modifying the algorithm.

Once the preprocessing module of the algorithm is modified as explained above, another modification is done to the Step S-2, i.e. the search for the most similar pattern. Instead of minimizing the distance between \( \text{dev}_T(u) \) and \( \text{pat}_T^k \), the algorithm now minimizes,

\[
d^{1,2} \langle \cdot, \cdot \rangle = \omega \times d^1 \langle \text{dev}_T(u), \text{pat}_T^k \rangle + (1 - \omega) \times d^2 \langle \text{dev}_T^2(u), \text{pat}_T^{2k} \rangle
\]

i.e., the summation of two distances where \( \text{dev}_T^2(u) \) denotes the secondary data event obtained from the secondary training image, \( \text{pat}_T^{2k} \) is a secondary pattern and \( \omega \) is a weight is attached to the combined summation to let the user of the algorithm give more weight to either the primary or the secondary values, reflecting the ‘trust’ of the user to the secondary data.

The net result of the above modifications is that, for every node \( u \), the algorithm now finds the most similar pattern not only based on the previously calculated nodes but also based on the secondary data. When there is also primary data available, this minimization is performed only after the patterns that condition to the primary data are found as explained in the previous section, i.e. primary data has priority over secondary data.
Examples

In this section, the results of several runs performed using the SIMPAT algorithm are discussed. Fig. 1a is an example of wavy bed lamination. Here, the training image contains complex patterns in the finest scale. Yet, higher scale patterns are fairly repetitive and thus can be considered 'simple'. A template size of $11 \times 11 \times 7$ is chosen to reflect the complexity of the fine scale patterns. Fig. 1b shows the final unconditional SIMPAT realization obtained.

Fig. 1c is a 7 facies training image depicting a tidal channel system. Another notable property of Fig. 1c is that, the image is highly non-stationary, especially in the higher scales, where one facies appears only locally (The dominant facies which runs through the lower x-axis). As the unconditional SIMPAT realization of Fig. 1d illustrates, the algorithm successfully captures the non-stationary behavior of the training image, while also retaining the stationary facies relations.

Fig. 2a shows a synthetic reference case with 6 facies. Two different data sets are sampled from this reference case to test the primary conditioning capabilities of SIMPAT. The first, dense data set (Fig. 2b) is used with a train image that is highly representative of the reference (Fig. 2c). Using such a training image guarantees that the number of conflicting patterns is kept to a minimum during the simulation. Yet, in a real reservoir, one would expect a reasonable amount of conflict between the available data and the selected training image. Thus, the final SIMPAT realization (Fig. 2d) obtained for this case should be viewed as a theoretical check of the conditioning capabilities of SIMPAT. The second, sparse data set (taking only 25% of the data shown in Fig. 2b) is used with a training image that contains patterns which are more likely to conflict with the available data (Fig. 2e). In this case, the data dictates stacked channels whereas the training image only has isolated channels. Use of this training image and the sparser data set can be considered as a representative conditioning example in a real reservoir (Fig. 2f).

Fig. 3 demonstrates the application of secondary data conditioning using SIMPAT. In this case, secondary data is obtained by applying a seismic forward model $F^*$ to the binary reference case (Wu and Journel, 2004). The same model is applied to the training image to obtain the secondary training image. The final SIMPAT realization (Fig. 3f) conditions to secondary data relatively well but pattern reproduction is somewhat degraded as made evident by the disconnected channel pieces. This issue, along with possible solutions, is further discussed in Arpat (2004).

Conclusion

In this paper, an alternative, pattern-based reservoir characterization methodology, SIMPAT, is presented. The algorithm utilizes patterns as its main working unit and thus can be considered as a method between object-based and pixel-based modeling techniques.

The main conclusions of the paper are: (1) SIMPAT borrows heavily from computer science image processing algorithms and thus is less bounded by the stationary requirement of the probability-based geostatistical methods; (2) The same algorithm works equally well both on categorical (such as facies) and continuous variables (such as permeability and porosity); (3) A new multiple-grid approach where the scale relations are tightly integrated, allows the algorithm to accurately model highly complex geological scenarios; (4) Primary (hard) data conditioning works well when the training image used correctly describes the geological scenario. For optimally handling ‘wrong’ training images, additional research is required; (5) The proposed secondary (soft) data conditioning allows indirect integration of physical laws in to the algorithm. Additional research is required to better understand this new methodology.

The reader is referred to Caers and Arpat (2004) and Arpat (2004) for further details.
References


Figures

Figure 1 – Unconditional SIMPAT realizations. [b] is obtained using a 11×11×7 template and [d] is obtained using a 15×15×5 template. 3 multiple-grids were used for both realizations.
Figure 2 – Primary (hard) data conditioning using SIMPAT. [c] is used for realization [d] and [e] is used for [f]. Both [d] and [f] are obtained using a $11 \times 11 \times 5$ template and 3 multiple-grids.

Figure 3 – Secondary (soft) data conditioning using SIMPAT. [e] is obtained using a $11 \times 11 \times 3$ template and 3 multiple-grids for both primary and secondary images.