CONSTRUCTION OF HYBRID GEOSTATISTICAL MODELS COMBINING SURFACE BASED METHODS WITH OBJECT-BASED SIMULATION: USE OF FLOW DIRECTION AND DRAINAGE AREA

A REPORT SUBMITTED TO THE DEPARTMENT OF PETROLEUM ENGINEERING

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

By
Alejandro D. Leiva
June 2009

I certify that I have read this report and that in my opinion it is fully adequate, in scope and in quality, as partial fulfillment of the degree of Master of Science in Petroleum Engineering.

_____

Prof. Tapan Mukerji
(Principal Advisor)

Abstract

The use of hybrid techniques aims at constructing new models which enable us to overcome the existing limitations in the reproduction of curvilinear structures. Improvements on a deepwater turbidite reservoir model combining several primary modeling methods are presented. In the initial approach, multiple points geostatistics (MPS) plays an important role in simulating and conditioning the realizations.

The new approach overcomes the limitations found in the MPS usage by combining surface-based (SB) methods with object-based (OB) simulation. Representation of flow direction and calculation of upslope areas on the rectangular grid elevation model are used to determine the drainage basin and simulation area (SA). The SA is then used to relate the previously simulated surface with the current geological event being simulated. The procedure is based on representing flow direction as a single angle taken as the steepest downward slope on eight triangular facets centered at each grid point. For a given anchor point joining the channel to the lobe in the channel-lobe parameterization, we obtain the influence and dependence area in a sequential process. The contour of the newly generated SA along with the cumulative density functions (CDF) retrieved from a process-based model output are employed to introduce variability to the OB geobody simulation. This procedure is repeated as many times as lobes are to be simulated, only updating the current base topography.

The improved model emphasizes using more realistic geological rules, especially on lobe orientation and erosion caused during the deposition of the geobodies. The lobe erosion process is simulated in such a way that flow direction and special topographic features related to the flow erosional power are accounted for. Objective rejection rules are taken into account in the implementation of the model. This gives rise to an automatic and user

independent algorithm. This approach is potentially capable of representing the internal features of the geobodies.

With respect to the model conditioning, area expansion and a novel grid deformation approach are used to lead geobodies to the data location in such a way that the facies and thicknesses observed in core data are honored. Finally, a simple but efficient and geologically consistent property population algorithm is introduced.

Acknowledgments

I would like to thank my advisors Tapan Mukerji for patiently guiding me through my research. His insightful advice and encouraging support made this work possible.

I am very grateful to the Department of Energy Resources Engineering, in particular to the Stanford Center, for Reservoir Forecasting for the financial support and academic guidance.

Thank you to my friends Mehrdad, Mohammad, Cedric, Abhishek, Daniel, Gerardo and all the others I am forgetting to mention for showing me different ways of thinking, teaching me about their cultures and making my time at Stanford an amazing experience. A very special thanks to my officemates and friends, Mike and Zeid, for the priceless conversations we shared during the last two year. Many thanks to the ERE professors for sharing their knowledge and making my way of thinking much better structured. Special thanks to the ERE staff for doing a great job and making foreign students life much easier.

Most importantly, I thank my parents, Felicita and Carlos; brothers, Francisco and Carlos; and sister, Maria, for supporting me from home. Being away from them made me realize once more how important they are in my life. They have taught me many important lessons, and have always given me their unconditional love and support.

Table of Contents

List of Figures

Chapter 1

1 Introduction

The proper spatial characterization of subsurface properties that control the movement of reservoir fluids is a key aspect of the decision making process involved in reservoir managing. The property values and spatial distributions are considered as data by the reservoir simulator, hence the results obtained are fully dependent on the input data. Modeling important spatial features is a challenging area that has been explored in multiple ways.

Classic geostatistics deals with the problem using two-point correlations through the variogram. These algorithms are computationally efficient, but the realizations lack realism. Multiple-point algorithms such as SNESIM (Strebelle, 2002) and FILTERSIM (Zhang et al., 2006) can reproduce simple curvilinear features with relative success, but currently are far from being able to reproduce complex features such as the ones found in turbidite outcrops. Process-based methods generate very realistic realizations by solving the flow and sediment transport equations, but only one realization can be obtained using an initial set of parameters and boundary conditions, and the simulations cannot be conditioned to hard data, since the realizations are constructed forward in time. Besides, obtaining each realization is extremely expensive in terms of the computational time.

The approach presented here, based on the work of Michael et al., (2008), breaks down the problem of simulating complex geological features into a set of smaller problems that can be tackled using the most suitable approaches. The goals of hybrid models are to speed up the simulation process, generate multiple realizations for a fixed set of initial parameters, and be possible to condition to hard data. A workflow on how the hybrid approach should be addressed is proposed and the initial deepwater turbidite hybrid

model by Michael et al. (2008) is modified to account for new aspects proposed in the methodology. One of the key new aspects is that the changes brought about by the previously simulated topography to the simulation of the new lobe are incorporated in the modified model, using surfaced-based modeling. Channel directions and catchment areas are computed using the method proposed by Tarboton (1997), described later in section 2.3.

1.1 Hybrid Modeling Workflow

Figure 1-1 schematizes the methodology on how hybrid modeling approach should be faced for simulating specific depositional environments. First, we start off understanding the deposition environment and identifying the structures that have to be considered, these are the structures that play a role in the flow simulation. Once the geobodies that we want to model are identified, we continue with their parameterization. These parameters must fully describe the shape of the geobodies and be able to be obtained from any related source of data such as geologic studies, seismic data, or process-based simulation.

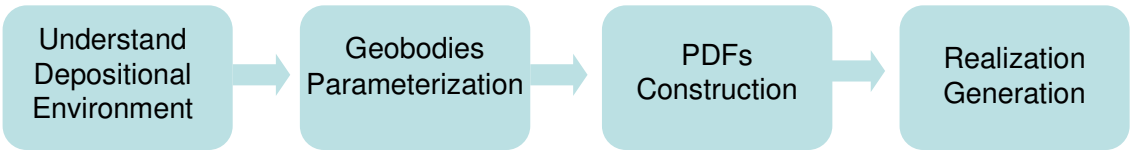


Figure 1-1: Hybrid modeling approach workflow

The generation of realizations follows a sequential process that starts with the simulation of major features (i.e., lobes, channels, etc.) and continues with the intermediate features (i.e., intermediate grained units, levees, etc.). The parameter values are drawn using Monte-Carlo simulation from the CDFs previously constructed. Then the two-dimensional realization (in most of the cases obtained using MPS algorithms or OB simulation) is post-processed to achieve the final thickness map which will be stacked on the current base surface. Consideration of geologic processes must be accounted for in

this part. Depositional layers will add thickness to the digital elevation model (DEM), whereas erosional processes will cut into the layers underneath in such a way that part of historical events will be removed. This sequential process will be repeated as many times as the number of major geological features we want the realization to have.

1.2 Deepwater Lobe Reservoir Workflow

Turbidite systems characterization has gained importance as off-shore exploration becomes more popular and feasible. Attempts to model complex turbidite systems using 2-point statistics have found difficulties in reproducing the complex features shown by outcrops and known to be present in turbidite systems. After 2-point statistics, OB techniques were the immediate approach to the problem (Haldorsen and Lake, 1984; Haldorsen and Chang, 1986). Recently, surface-based (Deutsch et al., 2001; Pyrcz and Deutsch, 2003; Pyrcz et al., 2005) and ruled-based techniques using MPS (Michael et al., 2008) have been introduced, but there has not been any approach that combines these techniques in a realistic and efficient way, in terms of computing speed (for the conditional case). The approach herein presented, based on work of Michael et al., (2008), uses surface-based simulation to determine feasible areas on which the OB simulation of geobodies is performed. The usage of OB simulation aims at obtaining an algorithm able to generate realization in a fraction of the time required by process-based approach, reproduce curvilinear features, and possible to be conditioned to hard data by using grid deformation and area expansion algorithm.

Assuming the availability of CDFs for the parameters that describe the geometries of the geobodies we want to modeled, the combined surface-based and OB simulation approach to simulate turbidite lobes deposition (Figure 1-2) is as follows:

- Drawing of the anchor point location according to a Poisson process with a spatially variable intensity function (Lantuejoul, 2002). The so-called P-field

(intensity function) changes after every major sedimentation event (lobe deposition).

- Computation of the influence area corresponding to the anchor point previously drawn from P-field using the infinite directions algorithm (Tarboton, 1997).

- Calculation of the dependence area relative to the points on the drainage flow path from the anchor point respecting the infinite directions algorithm.

- Determination of angle range of plausible geobodies orientation considering the furthest n-percentile of points on the boundaries of the initial simulation area (dependence area) to the anchor point.

- Construction of object-based realization of the lobe considering the variability incorporated by using randomly drawn parameters from the CDFs available and the valid angle range previously obtained (multiple object based realizations might be used subsequently as training images in case that MPS is to be used as proposed by Michael et al., 2008).

- In the case hard data are available, the simulation area and the OB simulation are constrained to these data. Layer thickness and facies spatial locations are honored.

- 2D to 3D transform of OB geobody simulation (one thickness and one erosion map for each accepted OB geobody).

- Stacking of the thickness and erosional maps on the base surface previously used in the influence area calculation.

- Presence or absence and thickness simulation of intermediate fine-grained unit.

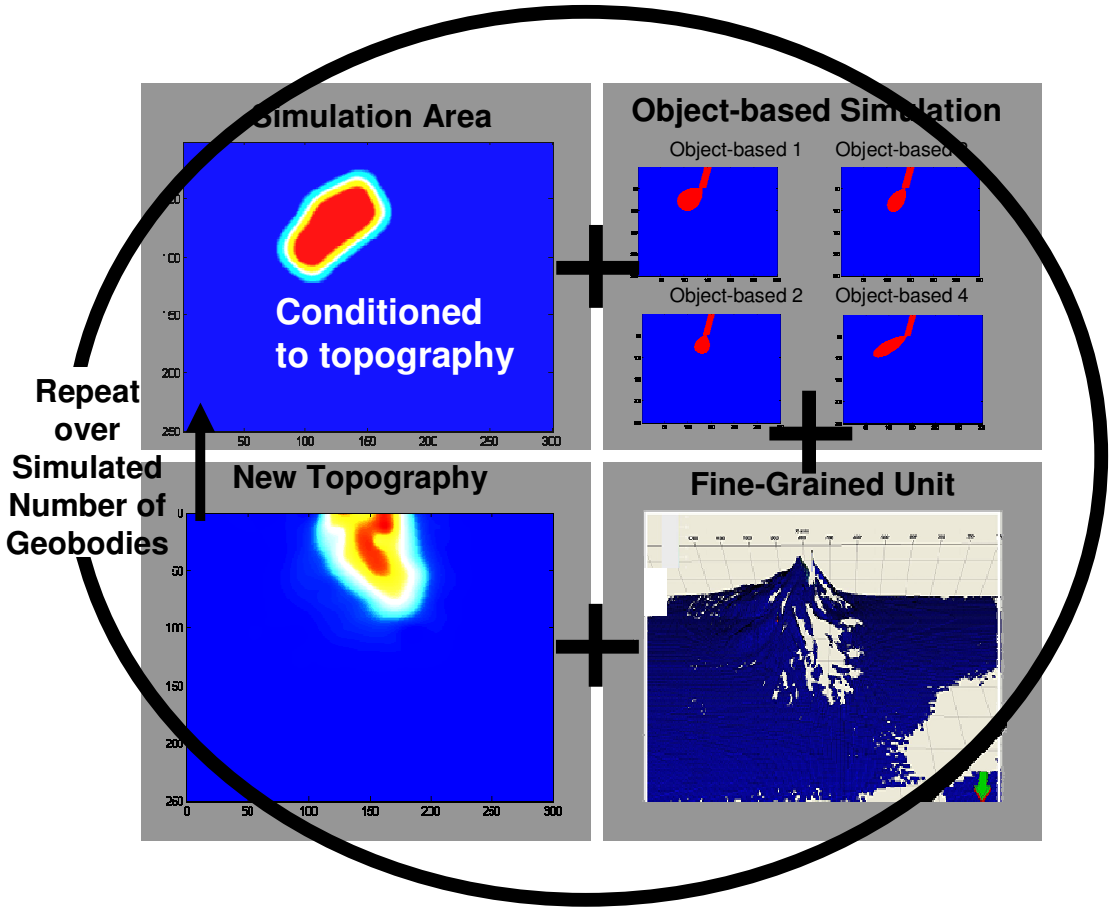


Figure 1-2: Deepwater Lobe Reservoir Workflow (modified from Michael et al., 2008)

Chapter 2

2 Model Construction

2.1 Construction of CDFs from Process-based Model

The statistics required to build the CDFs of parameters describing major features were obtained from the output of a process-based turbidity model by Tao Sun (left picture in Figure 2-1). After analyzing the model, it was possible to infer that most of the important features in the flow simulation can be characterized as lobes and channels. These channels join the sediment source to the lobe anchor point (red star in Figure 2-1). Besides lobes and channels, the presence or absence of a thin fine-grained layer between lobe depositional events was modeled as well.

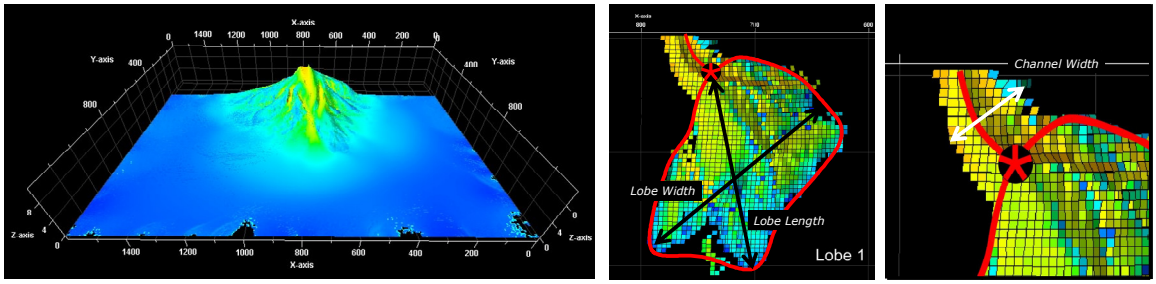


Figure 2-1: Process-based model parameters (Modified from Michael et al., 2008).

Lobe formation did not occur immediately after the underlying lobe deposition ended. Considering that, a CDF of the elapsed time after each lobe deposition was built. If this time was longer than a given threshold, the intermediate grained unit was modeled assuming a constant depositional rate per year; otherwise, the following lobe depositional event was modeled without considering a layer in between. It is really important to account for these thin layers since the grain size of particles that are contained in it are much smaller than the average of the model, converting them into flow barriers.

In contrast with the model proposed by Michael et. al. (2008), the methodology herein presented requires fewer parameters to be retrieved from the process-based model (only channel width, lobe width, lobe length and lobe maximum thickness are used for a lobe simulation). This is due the fact that our methodology omits the concept of progradation and migration in process of drawing the anchor point location, and just accounts for the sediment source location and previously simulated lobe location. This difference makes our methodology more generic, since any initial topography can be used without having to care about migration and progradation distance distribution anymore.

2.2 Lobe and Channel Parameterization

The two dimensional shape of the lobe is fully parameterized by the width and length of the geobody (lemniscate). Even though this parameterization does not capture the geobody shapes in detail, it is a good approximation of the representation of the geologic structures that most control fluid flow. The following equations describe the boundaries of the lobes:

$$r = a\cos(2\theta), x = r\cos(\theta), y = b\sin(\theta) \quad \textbf{(2-1)}$$

where a is the length, w is the lobe width, $b = c(w/a)$, and c relates length and width, usually within $[1,2]$. The angle θ is given by the lobe orientation which is related to the base topography by the relative positions of the drainage basin with respect to the anchor point. The channels are described by their initial points (sediment source), end points (anchor points), and channel widths (drawn from a CDF).

Certainly improvements on the parameter relationships can be achieved by knowing more about how these parameters are related. A key factor is to account for relationships between lobe parameters and depositional power of geological events. Simple rules might

be applied with respect to it, such as 'large geological events would deposit a large amount of sediments and therefore the channel width has to be proportional to the event magnitude' or 'lobe and channel width are intimately related, and these two are related at the same time to the topography slope, hence assuming a constant channel width is quite a big simplification'.

2.3 Flow Direction Computation

Multiple algorithms have been developed for the calculation of flow directions on digital elevation models. Among the most popular and widely used is '8 flow directions' or D8 introduced by O'Callaghan and Mark (1984). The D8 approach has the disadvantages arising from the discretization of the flow direction in only 8 possible directions, separated by 45 degrees. Fairfield and Leymarie (1991) try to overcome this problem by adding a random component and assigning a flow direction to one of the neighbors downslope. We picked out the so-called 'Infinite Directions' algorithm by Tarboton (1997) because of its capability to minimize dispersion (the flow coming from one grid is split up into at most two downslope neighbors), its simple and efficient grid based matrix storage, and its robustness in terms of handling irregularities in the DEM.

If we consider a single triangular facet (Figure 2-2) delimited by e_1, e_2, and e_3 (e_i and d_i are elevations and distances between pixels as labeled in Figure 2-2), the slope vectors S_1 and S_2 are given by the following expression:

$$\left.\begin{aligned} S_1 &= (e_0 - e_1)/d_1 \\ S_2 &= (e_0 - e_2)/d_2 \end{aligned}\right\} \quad \begin{aligned} r &= \tan^{-1}\left(\frac{S_2}{S_1}\right) \\ S &= \sqrt{S_1^2 + S_2^2} \end{aligned} \qquad \textbf{(2-2)}$$

where *r* and *S* are the magnitude and slope direction of the flow respectively. If *r* is not in the range $[0, \tan^{-1}(d_2/ d_1)]$, then *r* has to be set as the direction along the appropriate edge and S assigned as the slope along that edge.

$$
\begin{aligned}
&if\ \ r \notin \left(0, \tan^{-1}\left(d_2 \middle/ d_1\right)\right)\ then \\
&\quad if\ \ r < 0 \\
&\qquad r = 0 \\
&\qquad S = S_1 \\
&\quad if\ \ r > \tan^{-1}\left(d_2 \middle/ d_1\right) \\
&\qquad r = \tan^{-1}\left(d_2 \middle/ d_1\right) \\
&\qquad S = \left(e_0 - e_2\right) \middle/ \sqrt{d_1^2 + d_1^2}
\end{aligned}
\qquad \textbf{(2-3)}
$$

The eight possible values for the eight facets depicted in Figure 2-2 are computed. The local angle associated with the largest downwards slope is selected and adjusted to reflect an angle counter-wise from east.

$$
\begin{aligned}
&\rightarrow the\ local\ angle\ with\ \ the\ largest\ downwards\ slope \\
&r_g = a_f r^{'} + a_c \tfrac{\pi}{2}
\end{aligned}
\qquad \textbf{(2-4)}
$$

The multiplier a_f and constant a_c depend on the facet selected. Table 2-1 shows the a_f and a_c values for the eight facets.

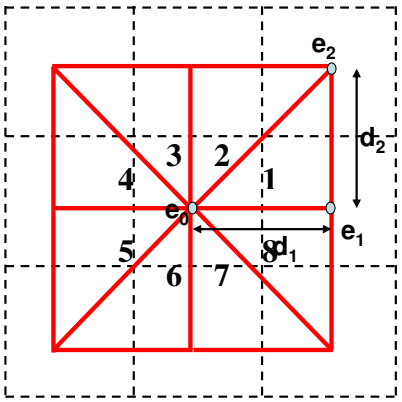


Figure 2-2: Eight Triangular Facets Layout.

Table 2-1: Statistic of non-conditional simulation.

Facet	1	2	3	4	5	6	7	8
e0	ei,,j	ei,,j	ei,,j	ei,,j	ei,,j	ei,,j	ei,,j	ei,,j
e1	ei,,j+1	ei-1,,j	ei-1,,j	ei,,j-1	ei,,j-1	ei+1,,j	ei+1,,j	ei,,j+1
e2	ei-1,,j+1	ei-1,,j+1	ei-1,,j-1	ei-1,,j-1	ei+1,,j-1	ei+1,,j-1	ei+1,,j+1	ei+1,,j+1
ac	0	1	1	2	2	3	3	4
af	1	-1	1	-1	1	-1	1	-1

2.4 Drainage Basin

The drainage basin is computed using the upslope area calculation procedure proposed by Tarboton (1997). The calculation follows a recursive method that is an extension of the very efficient recursive algorithm for single direction (Mark, 1988). The infinite directions algorithm splits the flow from one grid-block into up to two downwards neighbors with the largest downslope. The upslope area from a particular cell will be given by its own area plus the upslope area of the upslope neighbors that have some fraction draining to the pixel for which the slope area is being calculated. This recursive algorithm has the particularity of being extremely fast in terms of computational time, which makes it suitable for being used in a iterative way in case that the conditioning of the model requires it.

The points that are in the path that a droplet would follow as it drains down from the anchor point respecting the infinite directions algorithm constitute the influence area (IA).

We obtain the dependence area (DA) by calculating the upslope area associated to all the points in the IA (Figure 2-3). Below is the pseudo-code of the logic behind the infinite direction algorithm:

$$
\begin{aligned}
&Procedure\ DPAREA(i,\ j)\\
&\ \ if\ \ Area(i,\ j)\ is\ known\\
&\ \ \ \ then\\
&\ \ \ \ \ \ no\ action\\
&\ \ \ \ else\\
&\ \ \ \ \ \ Area(i,\ j) = 1\ \left(The\ area\ of\ a\ single\ pixel\right)\\
&\ \ \ \ \ \ for\ each\ neighbor\left(location\ in, ij\right)\\
&\ \ \ \ \ \ \ \ p = \ proportion\ of\ \ neighbor\left(in, ij\right)\ that\ drains\\
&\ \ \ \ \ \ \ \ \ \ \ \ to\ the\ pixel\left(i,\ j\right)\ based\ on\ angle\\
&\ \ \ \ \ \ \ \ if\ \left(p > 0\right)\ then\\
&\ \ \ \ \ \ \ \ \ \ call\ DPAREA\left(i,\ j\right)\\
&\ \ \ \ \ \ \ \ \ \ Area(i,\ j) = Area(in,\ jn) + p \times Area\left(in,\ jn\right)\\
&return
\end{aligned}
\qquad \textbf{(2-5)}
$$

The DA will delimit the lobe simulation (highlighted in green in Figure 2-3). The DA is used as SA, and its contour is utilized to incorporate variability in the object-based simulation. The OB simulations are stochastically built in such a way that they preserve the appearance of the geobody that wants to be represented, but having a high stochastic component.

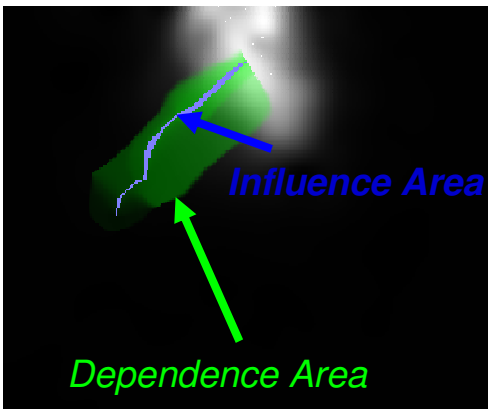


Figure 2-3: Influence and Dependence Area for a given anchor point.

2.5 Anchor Point Probability Field

The location of the anchor point is drawn according to a Poisson process with a spatially variable intensity function (P- field) previously generated. This P-field changes after each major depositional event. Since the lobe formation tends to occur close to the sediment source, for the first lobe depositional event, the P-field is constructed by considering a uniformly decreasing probability field starting from the sediment source. The probability intensity function becomes zero after the distance is larger than half of the longest dimension of the simulation.

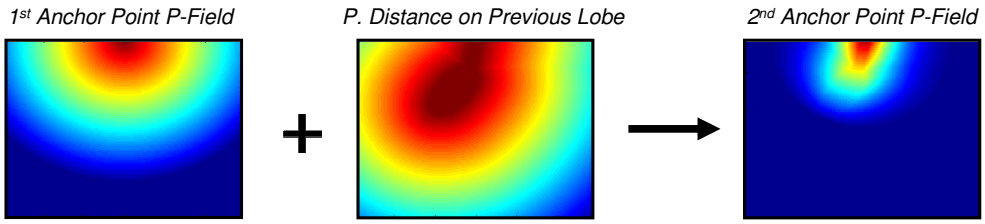


Figure 2-4: Anchor Point P-field dependence on previously simulated lobe.

It was observed in the process-based simulation output that after a lobe deposition the subsequent lobe tends to deposit close to the previous one, hence a P-field that decreases as the points location gets further from the previous MPS simulation area was computed. We obtained the joint intensity function, of the decreasing trend from the sediment source

P-field (used in the first lobe deposition) and the P-field constructed computing proximity distance on the previous lobe simulation area, using the *Tau Model* with parameters one for each P-field. After each lobe deposition, the P-field relating the location of the last lobe simulated is computed and then combined with the uniformly decreasing probability field to obtain the new probability field.

2.6 Lobe Orientation

The lobe orientation is related to the drainage basin. Each anchor point defines an influence and dependence area (Figure 2-5). The dependence area is considered as the drainage basin for that particular anchor point. The valid range of angle θ used for the objected-based simulation is defined by joining the anchor point to the furthest n-percentile (usually n around 30%) of the points on the drainage basin boundary. This angle range gives rise to the orientation of the lobes, which might be used as TIs in case the MPS approach is chosen.

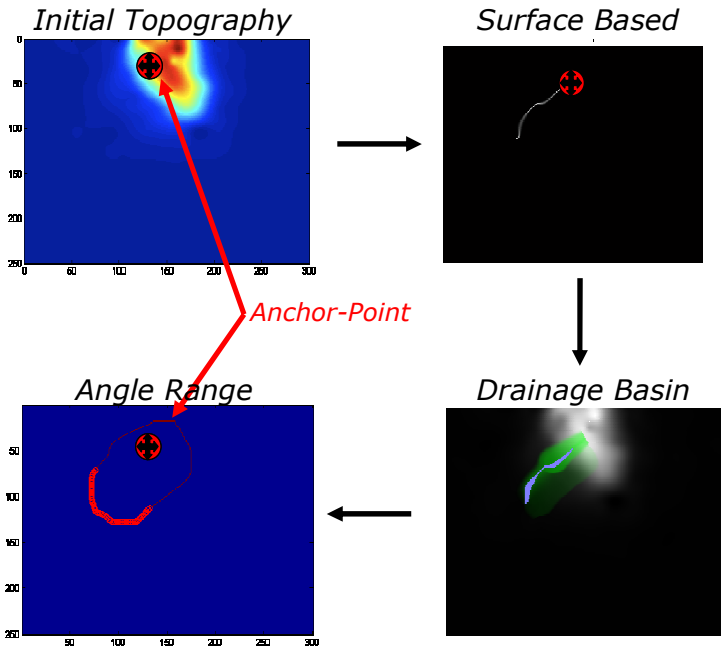


Figure 2-5: Possible angle range orientation based on drainage basin.

Figure 2-6 shows how the lobe directions in the TIs are related to the surface. This is required since a lobe deposition would follow the same orientation that a flow draining down from the anchor point would follow.

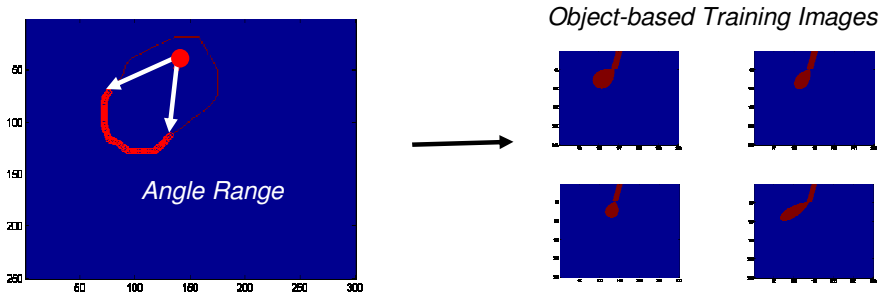


Figure 2-6: Object-based TI generation.

2.7 Simulation Area

In the case of combining surface based methods with OB simulation, the final simulation area is given by the drainage basin. On the other hand, when MPS is to be applied, the final simulation area is obtained by combining the initial simulation area (drainage basin) with the previously simulated TIs for the lobe currently being simulated. In the case of the channel zone (distance from the sediment source to the anchor point), since the drainage area does not considers it, the union of all the channels part of the TIs are taken. With respect to the position where the lobes hypothetically are, the intersection of the initial SA with the lobe part of the TIs built accounting for drainage basin separately is taken.

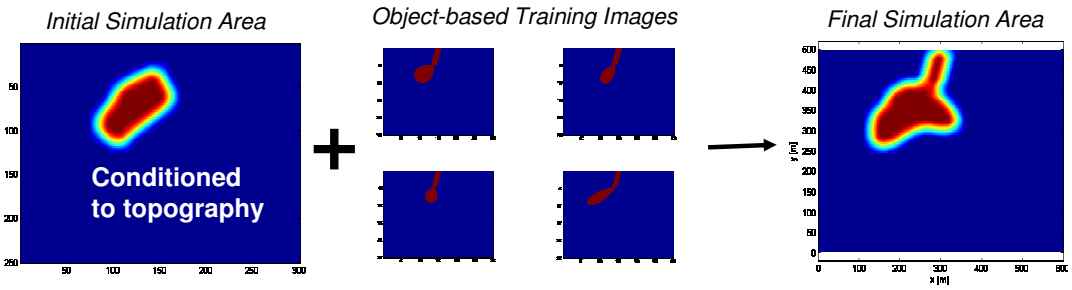


Figure 2-7: Final MPS Simulation Area.

2.8 2D to 3D Transform of MPS realization

The 2D to 3D transform is carried out by applying a distance transform on the binary image of the geobody simulation. Different distance transforms can be used in this respect. Proximity transform is essentially a distance transform followed by an inverse normalization of the resulting distances such that all the nodes of the transformed image range between [0, 1], with 0 indicating the furthest node and 1 indicating a target object node (Arpat, 2005). In our case, we use the inverse of this distance transform twice on the binary image. The proximity distance maps obtained therefore, have values in the [0, 1] range. For the first distance map, a low number indicates that the internal point is close to the 'boundaries' of the geobody. For the second distance map, the distance is computed using the anchor point as a center for the distance calculation (distance maps in Figure 2-8). The altitude values in the second distance map do not go to low values in the range [0, 1] since the values out of the lobe boundaries are set to zero. Subsequently these two distance maps are combined to generate a [0, 1] altitude map that mimics the altitude distribution observed in the process-based model.

The proximity transform brings some unwanted features to the [0, 1] valued image, such as shortening in the altitude values in the channel zone. Therefore, a compensation of low zones must be performed. Basically, it consists of increasing the proximity transformed values below a given threshold. Values lower than the threshold are augmented so that the geologic structures that they are meant to represent are preserved in comparison to the highest point in the altitude map. Then, the [0,1] ranged image is smoothened out using a local varying mean algorithm, consequently the possible artifacts due to the low zones compensation are removed. Finally, the [0, 1] altitude map is multiplied by the maximum thickness value drawn from a CDF constructed with information obtained from the process-based model.

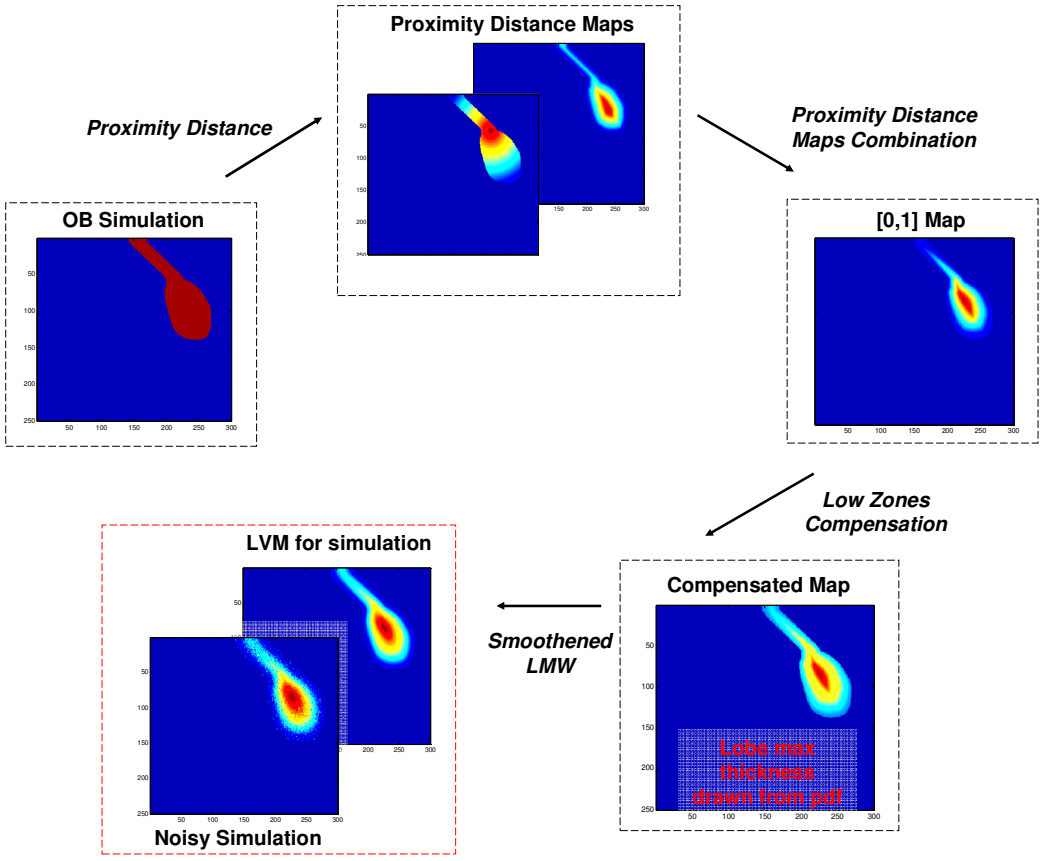


Figure 2-8: Thicknesses obtaining from OB simulation.

In the case of conditioning the simulation to hard data, the altitude values can be used as a local varying mean for a two-point geostatistics algorithm. Grid deformation and area expansion are able to use the original OB simulation. When there are no hard data available, adding random but geologically sound (locally correlated) noise to the OB simulation increases the realism of the altitude map. In addition, different forms of the proximity distance transforms can be used. Another good approach, given the availability of an underlying topography, would be the use of geodesic distance transform for gray-scale images as suggested by Toivanen (1996).

2.9 Erosion

Since the erosion is considered as a consequence of the depositional event (lobes erode the surface on their way downhill), it is simulated in such a way that only the topography

underneath the lobe is eroded. This way, there is no erosion where there is not a major depositional event above. The erosion process is simulated accounting for flow direction and special topographic features related to the flow erosion power. Topographic gradient and curvature are used to give erosion values at a given point in the topography under a lobe deposition. Locations with high gradient magnitude will be eroded more than the ones with low gradients, since the flow energy is assumed to be lower at locations with low gradients. With respect to the curvature profile, it is used in such a way that positive curvature (bowl shaped) indicates less erosional power than negative curvature, point at which the flow collide into surface barriers (dome shaped). The flow direction given by the direction of the flow in the influence area is considered in an 8-direction scheme (using infinite directions is impractical since the orientation of the channel along its whole length is described by an average direction). The same scheme is utilized to compute the gradient direction at any given location. At locations where the gradient direction coincides with the direction of the flow, under an 8-direction scheme, more erosional power is taken into account. The relative values will depend on the fluid properties such as viscosity, grain content, flow mass, etc. In this study, arbitrary weights are taken for the different aspects considered in the erosion, although further corrections to erosion-fluid properties are known to be required to obtain a more precise depiction of what really occurs in a geologic depositional event.

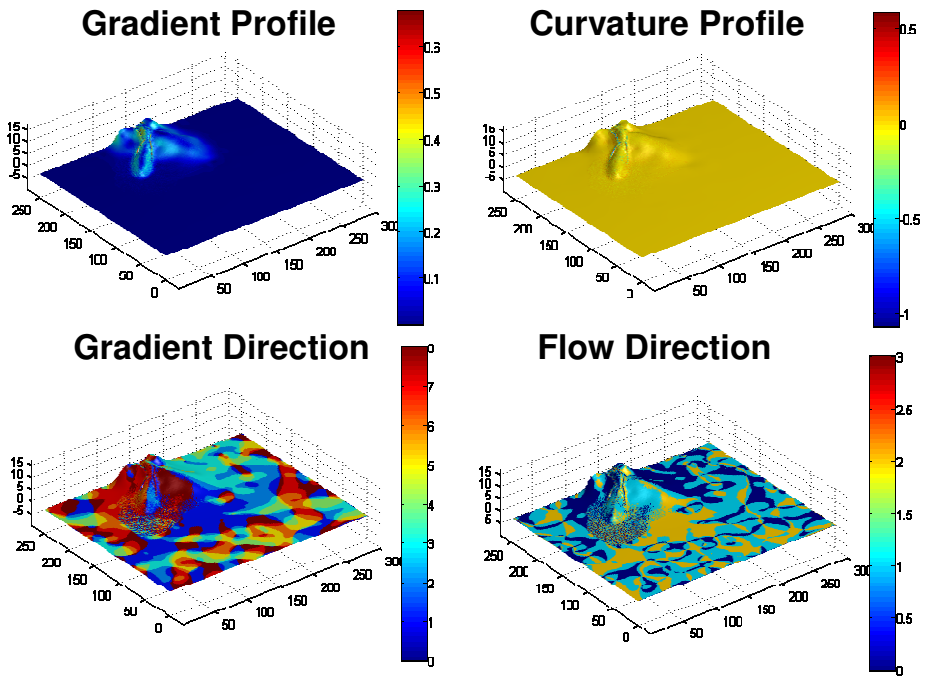


Figure 2-9: Flow directions and curvature and gradient profiles used in simulating erosion.

Figure 2-9 shows how the erosion map is different from zero only in the zone underneath the lobe deposited. In general, since the sediment source is assumed to be in locations where the base topography is high and shows irregularities, more discontinuities are expected close to this point in the erosion map as compared to locations far away from the sediment source (channel starting point).

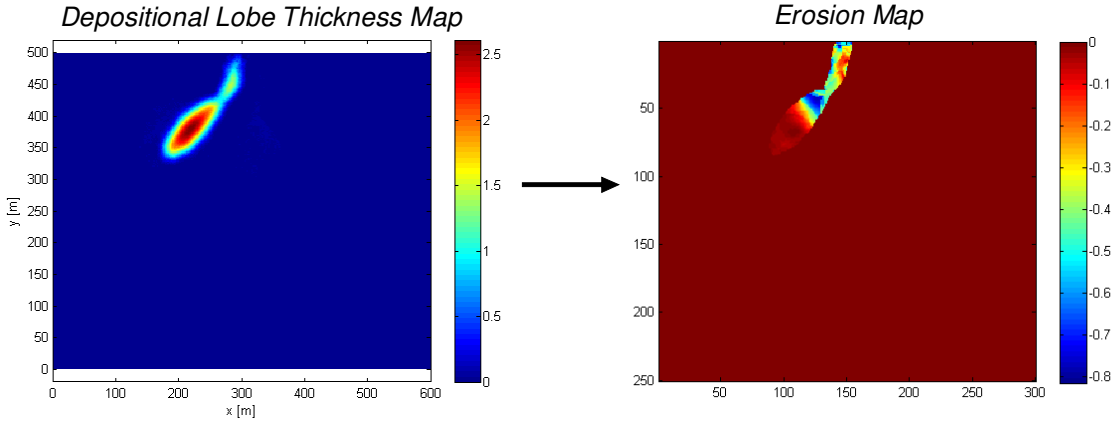


Figure 2-10: Depositional and erosion maps for lobe-shaped geobody.

By taking into consideration flow direction, gradient and curvature values, very realistic features present in erosional process, similar to the ones observed in landslides, are reproduced in the model. Figure 2-11 shows a close-up view of the features obtained by considering erosion just underneath a lobe deposition.

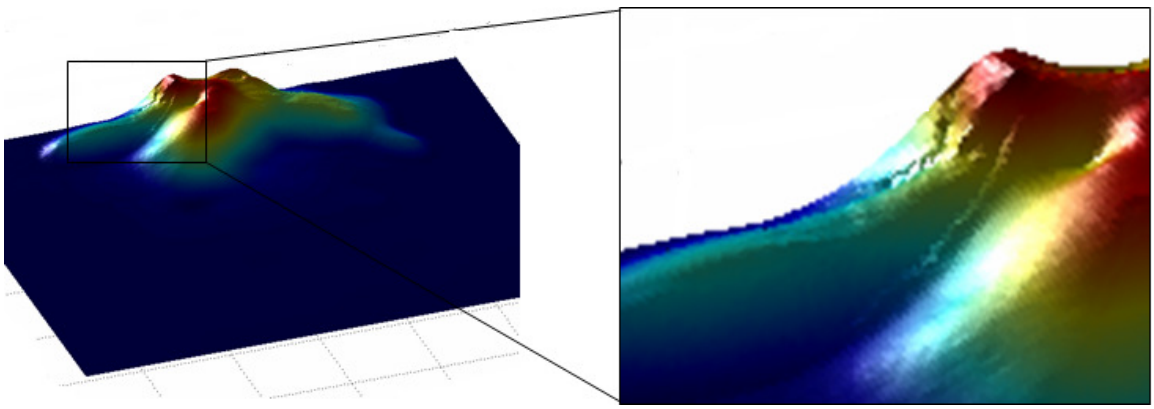


Figure 2-11: Erosion simulation underneath a lobe deposition.

2.10 Sediment Source Location

In contrast with previous works (Pyrcz et. al., 2005; Michael et. al, 2008), the sediment source position changes after each flow event. The position of the sediment source is always on the northernmost boundary of the model, whereas its east-west component is located in the range designated by the source windows (the altitude component is controlled by the base surface). This range is a user-defined parameter and is meant to avoid the over accumulation of sediment that occurs when a fixed sediment source location is used.

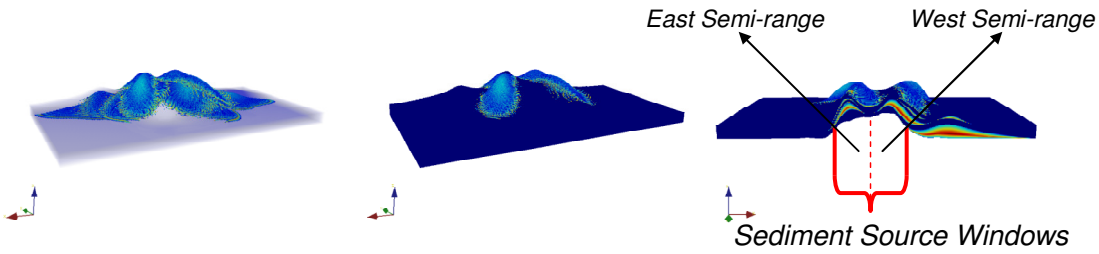


Figure 2-12: Sediment source location.

The exact position of the sediment source inside the sediment source windows (Figure 2-12) depends on the location of the furthermost point downhill in the influence area (IAFP) and the current topography in the valid range of x-coordinates for the source. The location of the IAFP determines on which semi-range of the sediment source windows will be the source, whereas the topography on the specific semi-range determines its exact position, which is given by the topographically lowest point in that particular semi-range.

A more realistic sediment distribution is achieved by using a location range for the sediment source position. Moreover, a better association between source and main geobody is accomplished since the both main features are always located on the same semi-simulation space.

2.11 Rejection Rules

The rejection rules are almost as important as the geologic rules used in the hybrid approach. They are meant to ensure that the model realization is comprised of geobodies physically feasible and geologically realistic. These rejection rules are extremely useful in case MPS is used (since MPS suffer from the drawback of generating unconnected geobodies very frequently) and must be strict but also must let the algorithm to run within a reasonable amount of time to make it really competitive with process-based simulation, which in general are known to run over extremely long time.

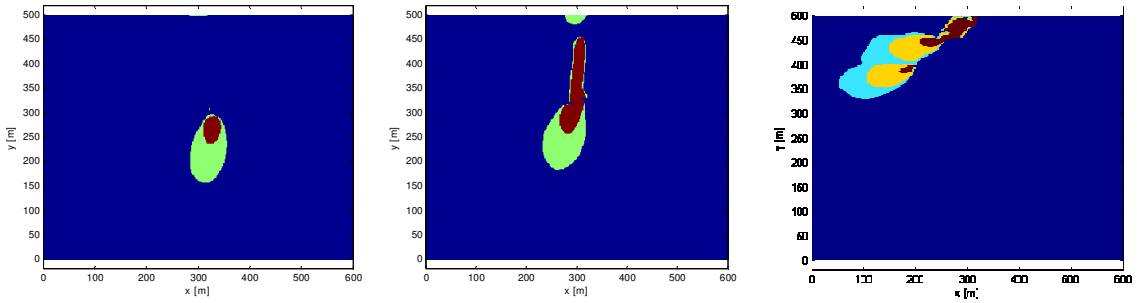


Figure 2-13: Examples of realizations that should be rejected.

Some examples of rejection rules already implemented in the simulation code are shown in Figure 2-13 (in this example we consider the sediment source to be in the center of the northernmost boundary). Most of these conditions are imposed on the MPS simulation. We require the simulated lobe to be connected to the sediment source. According to this, Figure 2-13 on the left should be rejected. The MPS simulation must be comprised of just one geobody. Fig. 2-13 in the center shows an example when this condition is not met, whereas the Figure on the right is a feasible lobe simulation. All the not-allowed lobe simulations are possible outputs of any MPS algorithm that respects the proportions of each facies required (in this case lobe/no lobe facies), hence it is necessary to check each realization. Additionally, conditions are imposed on the direction and length of the channel that determines the dependence area. In general it is required that the channel flows away from the sediment source; but this is not always achieved directly since the topography changes without any constraint, rather only accounting for the depositional events.

Chapter 3

3 Model Conditioning

Despite being constructed utilizing the pseudo-discrete step given by each geobody deposition, the algorithm herein proposed has the capability of being conditioned to hard data. On the other hand, process-based algorithms can simulate geologic events very realistically, but since they are built further in time, making conditional realizations is extremely difficult. Also, just one realization can be obtained from a fixed set initial of parameters (typically there are no stochastic processes involved in the generation of realizations), which certainly is a disadvantage compared to other techniques (MPS, OB, etc...).

Some possibilities for conditioning the model to hard data are as follows:

- Data conditioning layer thicknesses.
- Data related to petrophysical properties distribution.
- Data related to layer interfaces and type of contact (erosional or depositional).

When the lobe simulation is to be led in a particular direction, iterations on the influence and dependence area are necessary. The iterative process will stop when the simulation area contains the location where the hard data are located. All this calculation turns out to be very fast since points belonging to previous drainage areas are excluded from the allowable anchor-points in the next iteration. This is due the fact that the probability of a point of being in two dependence areas obtained from two different influence areas is zero (Figure 2-3). In case OB simulations are used as TIs for a model that uses MPS to make the realization conditioned to hard data (Michael et. al., 2008), taking advantage of the usage of flow directions and drainage area is as easy as setting the simulation area for the conditional simulation as the drainage area previously calculated. More constraints

can be applied on this SA as explained in section 2.7 of this report. With respect to the surface based methods with object-based simulation approach, two different procedures on how to approach the conditioning are developed in this work: area extension and grid deformation. The advantages of using these approaches are in the computing time, which is reduced considerably compared to realizations conditioned using MPS, and the adoption of hybrid techniques that account for underneath topography, making the turbidite system model a more accurate representation of an actual geological event.

Simpler conditioning algorithms such as the calculation of the spline residual map proposed by Pyrcz (2004) or thicknesses obtained using 2-point geostatistics algorithms (SGSIM, direct sequential simulation, etc.) considering an altitude map previously computed that serves as a local varying mean can be applied. However these algorithms trim the lobe surface without any particular concern on how geologically accurate this action is, therefore their realizations lack realism and many times can become depictions of an unfeasible geological event.

3.1 Area Expansion

This particular way of conditioning the model to well data (facies and layer thickness) is applied in cases when the hard data are located in the area delimited by the OB simulation, but the condition of maximum thickness is not meet, when the algorithm proposed in section 2.8 to compute the altitude map is used. It is required to have the hard data in the OB area since the possible perturbations on the altitude map that can be obtained by this methodology are not very large (grid deformation is used for more drastic perturbations). Therefore constraints have to be applied on the area expansion use. Essentially this approach looks for an intermediate base area between the OB simulation and the drainage basin that, after applying the distance transformation to compute the altitude map, matches the data at a given location, and also honors the maximum lobe thickness allowable.

Figure 3-1 shows an example on how after applying area expansion the area obtained matches the hard data and also the maximum lobe thickness constraint.

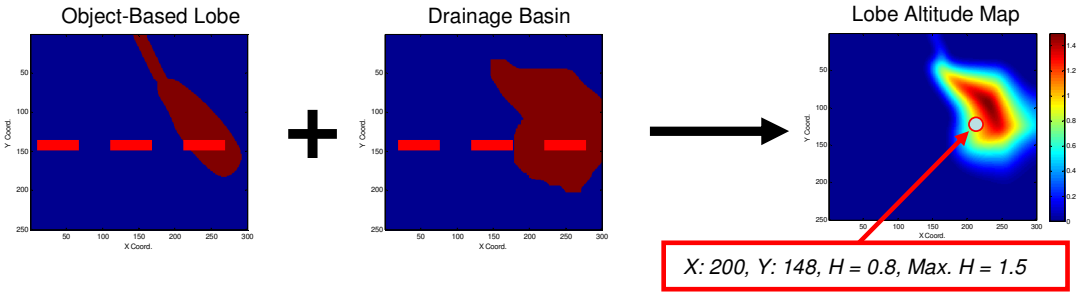


Figure 3-1: Drainage base and Object-based lobe used in area expansion.

Area expansion works as a simple iterative optimization algorithm which stops when the maximum lobe thickness is less or equal to the maximum lobe thickness allowed, which is defined by the user. Figure 3-2 shows the iterations required to meet the conditions shown in the Figure 3-1. Notice that every iteration meets the hard data thickness at location (200,148), whereas the maximum thickness (in this case set at 1.5) is not met until the last iteration. We achieve to match the altitude at the data location just by multiplying the [0, 1] map obtained after applying the distance transforms on the 2D image by the inverse of the altitude in the same map at that location times the length of the facies type observed in the well. By doing so, the desire geobody profile is always preserved, which as explained in section 2.8, mimics the altitude distribution detected in the process based model.

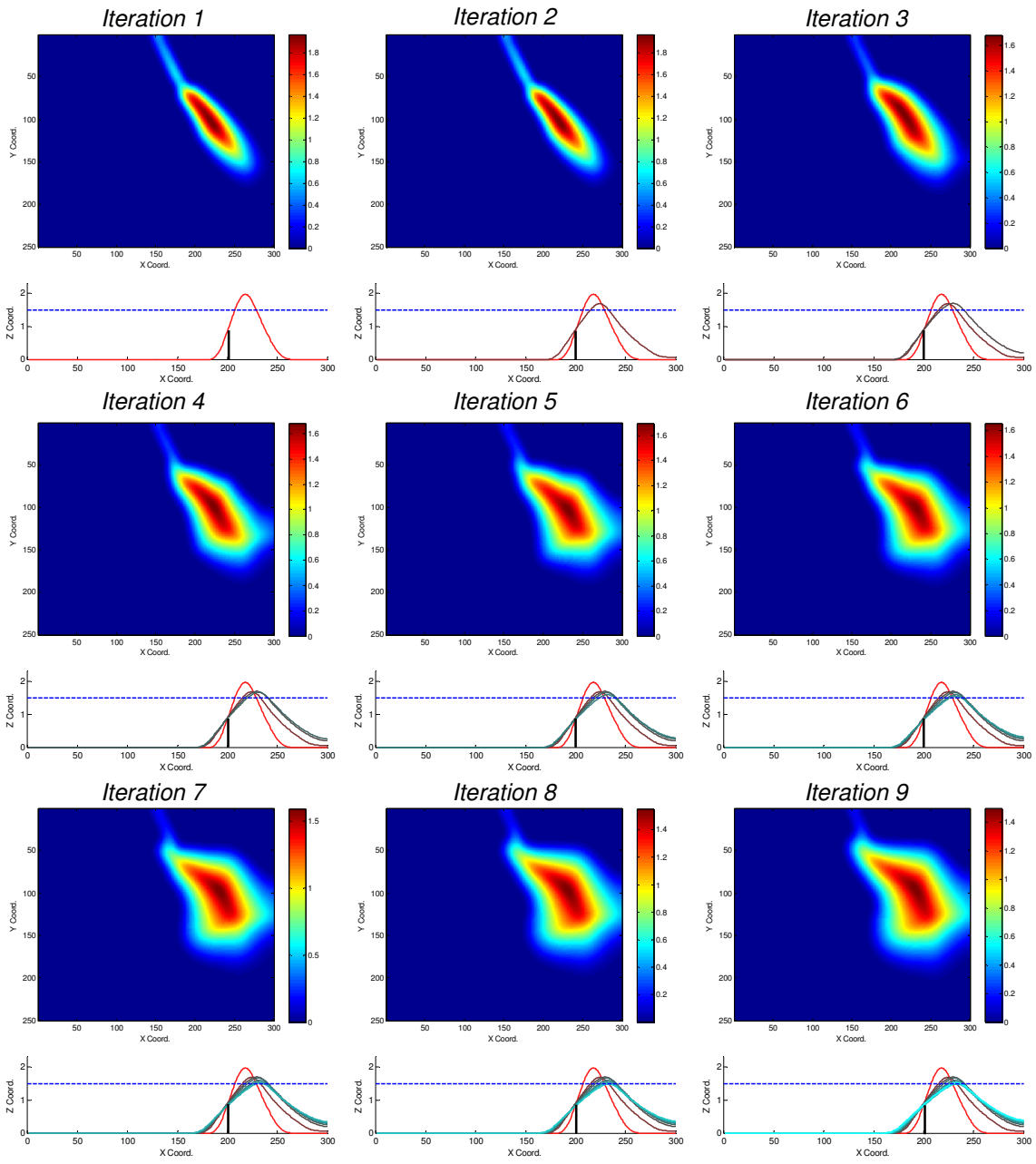


Figure 3-2: Area expansion iterations.

At every iteration the datum is matched (black line), but the maximum lobe altitude (dashed blue line) is only matched after 9 iterations.

3.2 Grid Deformation

Grid deformation as its name indicates works by deforming the X-Y grid on which the OB simulation is performed. This deformation is carried out in such a way that deformed geobody matches the data at a particular position. Figure 3-3 shows how the deformed

geobody looks after applying the deformation. Also how the constraint on the maximum lobe thickness is met can be seen in the same figure.

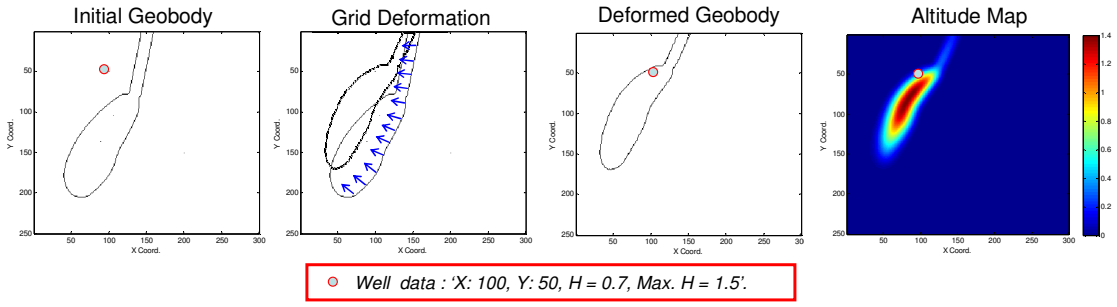


Figure 3-3: Grid deformation.

A displacement vector is decomposed in its X and Y components. The deformation of the grid is achieved by computing coordinate displacement independently for X and Y. The orientation of the displacement vector is defined by two previously computed vectors. The first one is given by the direction from the target point (well position) to the closest point on the boundary of the geobody. The second vector is orientated from the position previously computed on the boundaries of the geobody to the closest point in the highest 90 percent of altitude. The final displacement vector is given by adding these last two vectors.

Figure 3-4 exemplified how the displacement (ΔY) in the Y direction is computed for the initial and target point configuration shown. The component is placed at the location of the initial point and the four corner of the grid are set with zero as displacement. The last is done in order to avoid irregularities in the deformation.

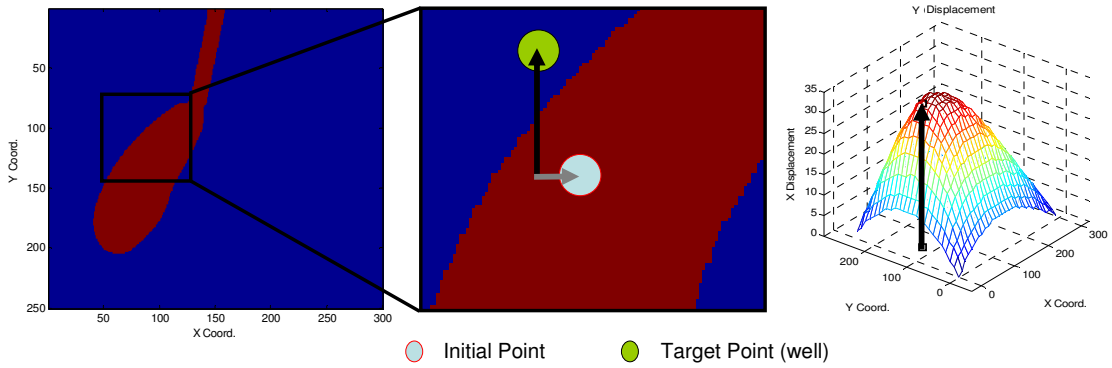


Figure 3-4: Grid deformation displacement computation.

The resulting binary image is given by translating the data placed at (X,Y) to (X',Y'), locations described by the following expressions:

$$\begin{aligned} Y' &= Y + \Delta Y \\ X' &= X + \Delta X \end{aligned} \qquad \textbf{(3-1)}$$

Each component is used independently as data, which is interpolated using any interpolation method (Triangle-based cubic interpolation is this case).

In the case that the new coordinates are out of the boundaries of the model, they are just set to be on the boundary. Some conditions can be applied on ΔY and ΔX to make the transition smoother, in case there is a zone where the new coordinates are out of the boundaries, but certainly they would not affect drastically the results.

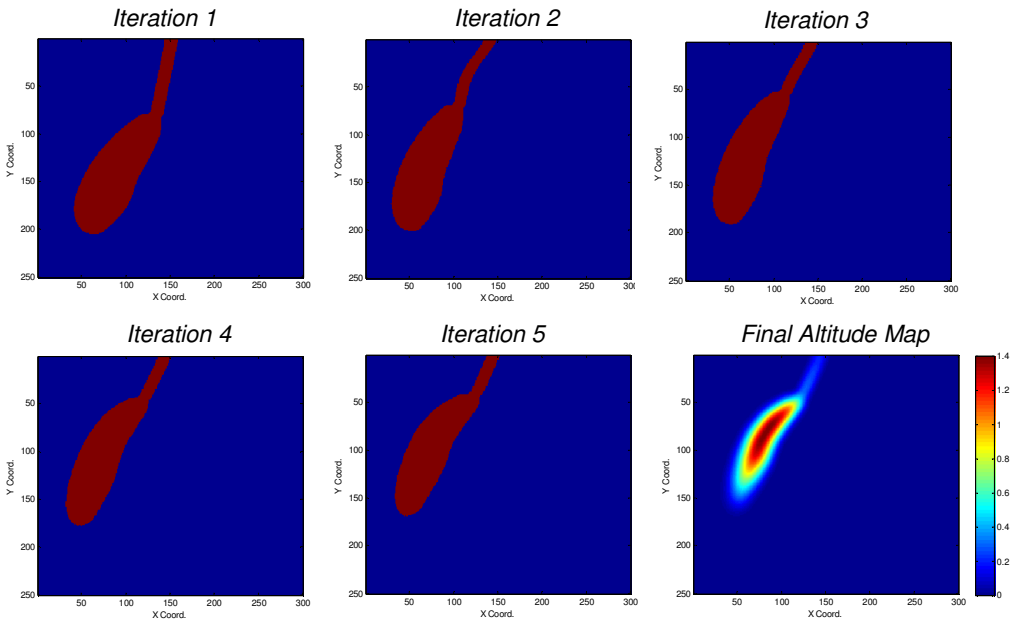


Figure 3-5: Grid deformation iterations until match hard data shown in Figure 3-3.

Deformation grid as area expansion is a gradual process that requires multiple iterations. Deformations are performed by increments along the displacement vector until finding the first fraction of the displacement vector that meets the constraints (max. lobe thickness and core length at the data location). The approach is iterative in order to perturb the minimally the geobody (Figure 3-5 shows an example of the iterations required to match the data). Also, by selecting as final point of the displacement vector a point in the highest 90 percent of altitude, it is ensured that the core length at a given position will be matched as long as it is less or equal to the 90 percent of the maximum lobe thickness defined by the user.

Even though it is possible to obtain very drastic changes on the original OB simulation by applying grid deformation (Figure 3-6), its use is not extended further than the region delimited by the drainage area. This constraint is taken in order to respect as much as possible the underlying topography, since larger displacement would make the simulation run faster, but would not be geologically consistent.

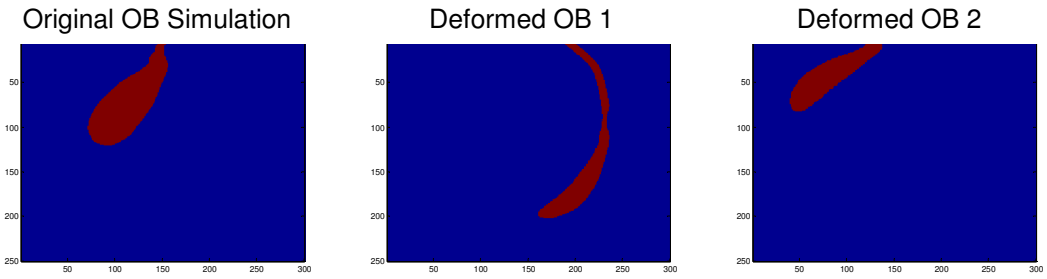


Figure 3-6: Examples of deformed geobodies.

3.3 Obtaining a Valid Conditional Simulation Area

Two different methodologies to match the data were presented when the well is located in either the OB simulation area or the drainage area. Grid deformation and area expansion are applied depending on the situation when the lobe candidate related areas (OB simulation and drainage area) happen to overlap the data locations. However, many times after a certain number of lobe depositions the data are still not matched. In this kind of situation it is required to condition the simulation area to the well location, since it is a requisite to have a depositional event in that particular position. The conditional simulation area is obtained by iterating on the drainage area used in the lobe orientation calculation. The iterative approach works as a smart process that accounts for the fact that points in a drainage area would never generate influence areas capable of giving rise to dependence areas out of the initial drainage area. This way, all the points belonging to a previous drainage area discarded from the candidates to become the following anchor point.

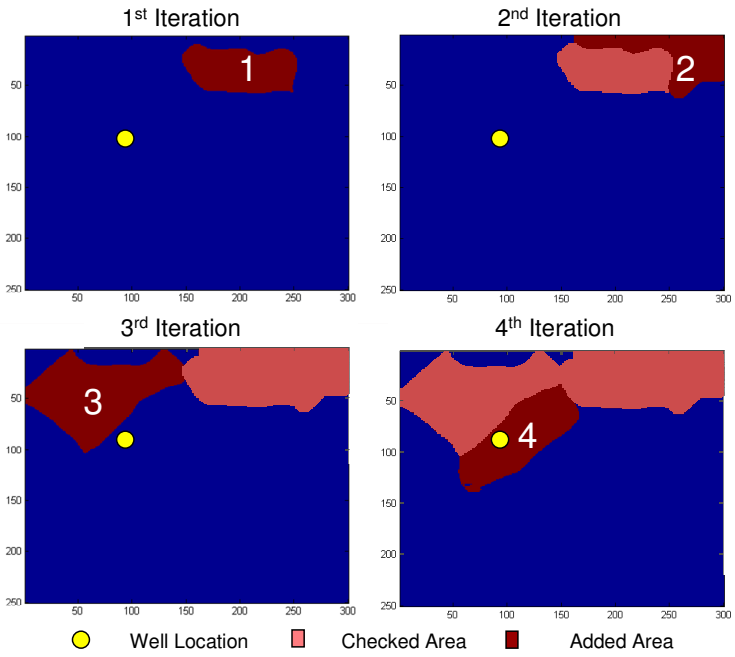


Figure 3-7: Simulation area iterations.

Figure 3-7 shows a simulation area searching process which ends when the well location (yellow dot) is overlapped by the drainage area. In this particular case the number of iterations were 4, but it might require a much larger number. The complexity of the problem will be given by the irregularities shown by the base topography. For irregular topographies, it is more likely to have to iterate longer to find the appropriate simulation area. In cases where the base topography shows irregularities that do not let the model to run in a reasonable amount of time, it is advisable to smooth out the surface before starting to iterate.

Chapter 4

4 Property Distribution

Once the reservoir geometry has been modeled accounting for the available data (lithofacies) and conceptual models, the next step is to model the porosity and permeability distribution inside the geobodies taking into consideration the trends observed in them. Xingquan et. al. (2008) assumed a distribution of progressively finer sediments from proximal to distal of the centerline of the OB geobody. The trend information is integrated into a cell-based facies model as locally variable facies proportions model. Xingquan et. al. apply Truncated Gaussian Simulation for rock types modeling, as it is able to capture the natural ordering of the facies type. On the other hand, our approach consists of generating a secondary variable that can be related to any other variable to be modeled that shows similar spatial distribution. This secondary variable considers the relative location of a grid cell in the geobodies and also the relationship between different geobodies (given by the maximum lobe thickness allowed) in the same reservoir model.

4.1 Secondary Variable Computation

The secondary data herein presented is a continuous variable for the sand type of facies. It can be considered as a level particles sorting inside the geobodies (high values indicate good sorting). Three possible ways of computing this secondary variable are described. The main difference among them is the amount of geometric information considered in the computation of the spatial correlation.

Figure 4-1 schematizes the three strategies proposed for the property distribution calculation. As we can see, only the gridblocks labeled as sand are assigned with this secondary variable. Something similar might be done in the case of shale facies, but given

the importance of high permeable path (sand paths) for the flow simulation, the proper property simulation is more relevant in high porosity-permeability zones.

Property distribution 1 only considers the relative position of a gridblock in a geobody. This is, the position with respect to the upper boundary of the lobe along the vertical. In this case, the secondary variable decreases its value linearly as the gridblock is located closer to the upper geobody boundary. There is no horizontal trend, and the same proportion of clean to dirty sand vertically is found along the whole geobody.

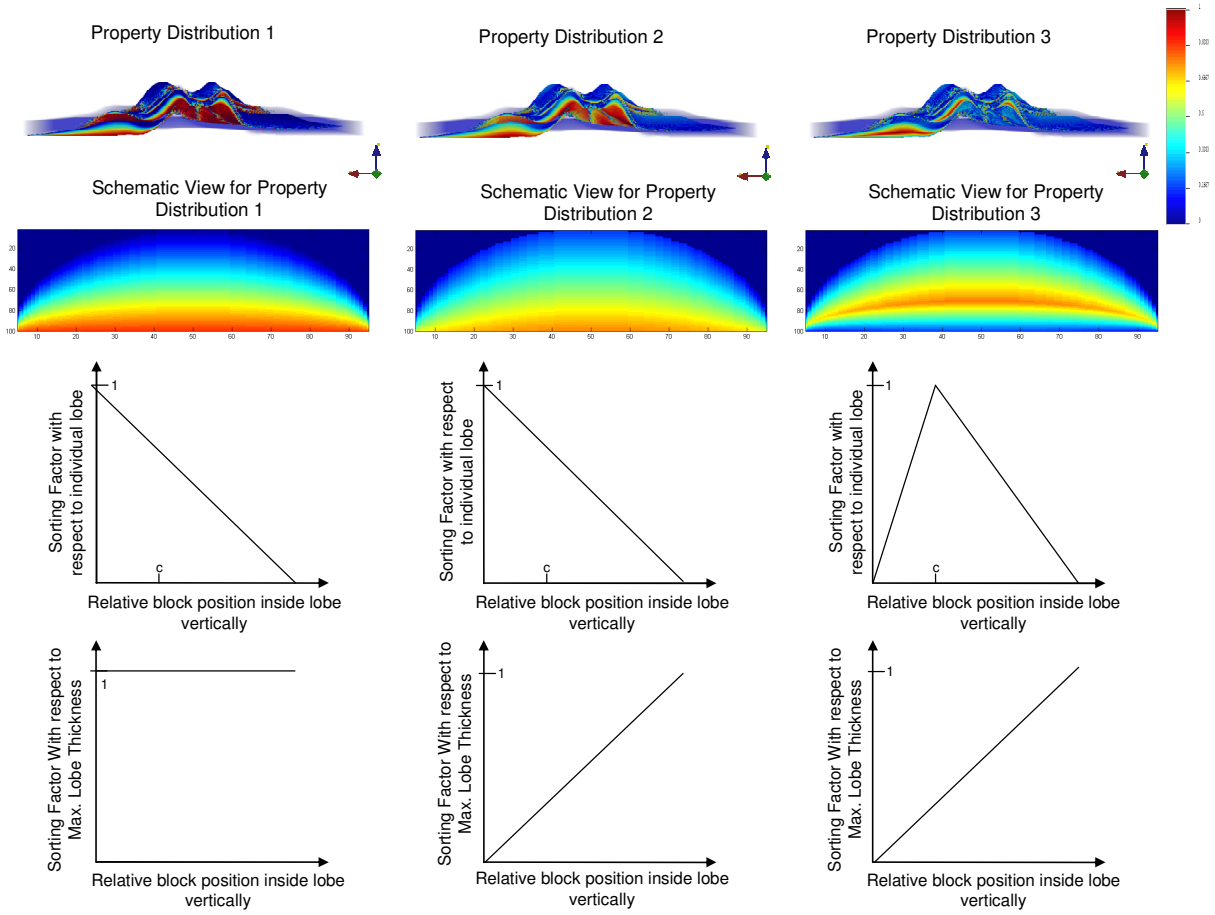


Figure 4-1: Different property distribution strategies.
The scale goes from zero to one according the sorting factor. High values denote well sorted grid blocks.

Property distribution 2 combines information of each lobe with the maximum lobe thickness allowed for the simulation. This way, there is less clean sands in the zones where the lobes pinch out. This property distribution is accomplished by assigning weights to the two components that comprise this particular secondary variable formulation. The first component relates the gridblock relative altitude in the geobody with the local thickness upper boundary of the geobody, whereas the second component relates the gridblock relative altitude with the maximum lobe thickness allowed by the user (see equation 4-1). By using a two component formulation, a horizontal trend is introduced. Also, as mentioned before, less clean sands are considered in thinnest part of the geobodies.

Finally, property distribution 3 uses an internal trend to avoid having linear decreases in the secondary variable as the cell is located upwards in the formation. Figure 4-1 shows that the best sorting level is reached at c in a normalized internal local lobe altitude. Using an internal trend allows us to mimic in a superior way the property spatial distribution, since this particle sorting spatial distribution resembles better what is seen in the process-based model output. Essentially, by using the trend, it is avoided to have the cleanest sands at the very bottom of the sand body, rather they are located at a user defined relative position. In this particular case, at c relative to the standardized [0, 1] local lobe thickness.

The expression used to compute the secondary variable $SV_{i,j,k}$ for the property distribution 3 is as follows:

$$SV_{i,j,k} = A_{i,j,k} \cdot (1-h) + Z_{i,j} \cdot h / mt$$
$$A_{i,j,k} = \begin{cases} S_{i,j,k} / c & , \ S_{i,j,k} \le c \\ (1-S_{i,j,k}) / (1-c) & S_{i,j,k} > c \end{cases} \quad \textbf{(4-1)}$$

where $S_{i,j,k}$ is the standardized cell position altitude with respect to the local lobe upper boundary $Z_{i,j}$, c the point of maximum level of sorting in the linear trend, mt the maximum lobe thickness allowed, h the weight given to the local altitude with respect to the maximum thickness, and the indexes i, j, and k refer to the block position in Cartesian coordinates.

Figure 4-2 displays the sand type of blocks corresponding to the secondary variable values above 0.1, 0.5, and 0.9 of an eight-geobodies simulation, which was populated using the 3 strategies previously described (only the blocks assigned with sand in the facies model were populated). We can see that the alternative 2 and 3 show a more effectively distribution of very clean sands (values above 0.9), whereas alternative 1 shows a very erratic distribution and an overestimated amount of blocks for the same category. Table 4-1 shows statistics on the three alternatives applied to the same reservoir model.

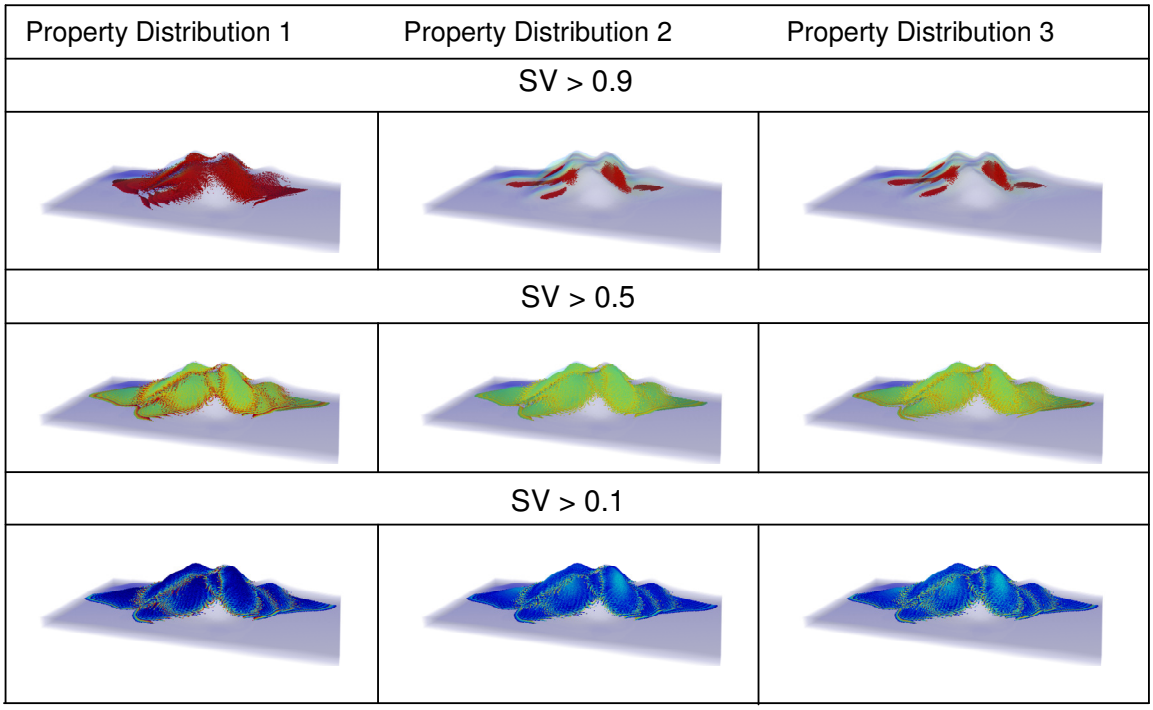


Figure 4-2: Reservoir blocks filtered according to secondary variable.

Amongst the three alternatives, 2 and 3 show very similar results, whereas alternative 1 is quite off from them, especially for high values of the secondary variable. By looking at Figure 4-2 we can say that alternative 3 shows the most realistic spatial distribution among the tree schemes, although its property distribution is quite similar to alternative 2 (more differences can be observed in the schematic view in Figure 4-1).

Table 4-1: Different Property Distribution Statistics

	Property Distribution 1	Property Distribution 2	Property Distribution 3
Prc. Blocks > 0.9	10.17%	1.61%	1.58%
Prc. Blocks > 0.5	50.13%	48.09%	47.69%
Prc. Blocks > 0.1	90.23%	97.30%	97.69%
Sand/shale ratio	0.3257	0.3257	0.3257
Sand percentage	24.57%	24.57%	24.57%
Shale percentage	75.43%	75.43%	75.43%

The internal trend used in this case is just and example, but many other trends might be used to simulate the secondary variable. Moreover, the assumption of a linear trend is quite arbitrary. Also, local dependency of the trends possibly relating to the underlying topography can be an interesting topic of research.

4.2 Porosity and Permeability

Porosity and permeability are continuous variables within each facies type. Porosity should be simulated independently within each facies type, and then the results merged on a per-facies basis. Linear regression among the porosity data positional dependent paired with the secondary variable can be used in order to find the correlation required for a two-point geostatistics algorithm that accounts for secondary data. The simulation of porosity using the appropriate algorithm will output property realizations that account for the geological knowledge of the geobodies shape. The latest assumption will be valid as long as the trends used in the secondary variable computation are geologically sound.

Permeability can be simulated with the corresponding porosity realization as secondary variable for collocated cokriging on the per-facies bases (Deutsch, 2002; Deutsch and Journel, 1998).

Chapter 5

5 Simulation Results

Besides the visual geological consistency of our model, a quantitative assessment of the most relevant input parameters defined by the user is presented in this chapter.

Although the original output of the model is a layer-based realization, the assessment of the parameters is performed on a regular grid-based model. A very intuitive but efficient technique for extracting a block-based model from a layer-based model is presented. This technique allows generating block-based models in different resolutions from the same layered model.

5.1 From a Layer-based Model to a Block-based Model

The methodology proposed for the construction of a grid-based model out of a continuous layer model consists of a vertical discretization of the domain taking into consideration the facies deposited in each layer. For a given Z-grid size, the process start from the bottom to the top of the model in incremental steps of the Z discretization size. If at a given location the closest layer upwards indicates sand (green layers in Figure 5-1 on the right), the sand flag is assigned to the block model. In case it is a red layer, the location would be assigned as shale.

The maximum resolution allowed in the plane $X-Y$ is given by the original gridding used in the combined surface-based technique, whereas in the case of Z direction, virtually there is no limitation (although a refinement that allows to preserve the relevant features for the flow simulation is enough). Certainly this is an advantage of this model as compare to other approaches used to generate reservoir models, which the time spent on

each realization generation is highly dependent on the resolution required (MPS, process-based simulation, etc..).

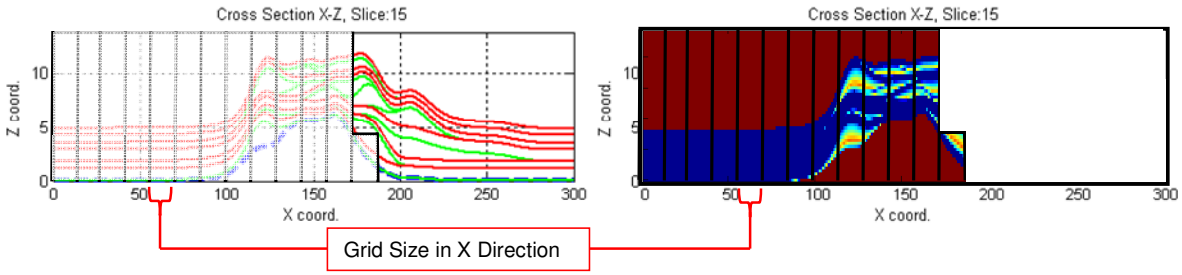


Figure 5-1: Gridblocks property assignment.

Figure 5-2 shows examples of discretization for 3 different vertical sections of an eight-lobe non-conditional simulation. This figure reflects some of the issues that arise due to the discretization. If the discretization is too coarse, many features are lost. Even more, given the discretization methodology, it might be possible that the facies assigned to the gridblock is not the most representative of the whole volume that it describes. Some kind of upcaling technique can be applied in order to avoid these artifacts. On the other hard, when the discretization is too fine, very highly detailed models are obtained. However these models are impractical in case flow simulation is to be run on them. The tradeoff between model resolution and model discretization finds its equilibrium at the resolution that preserves most of the geological features important in the flow simulation, but also that lets the simulators run in a reasonable amount of time. In the example shown below, certainly 100 blocks in the Z direction is an exaggeration since the same features can be captured with 50 blocks discretization in the Z direction.

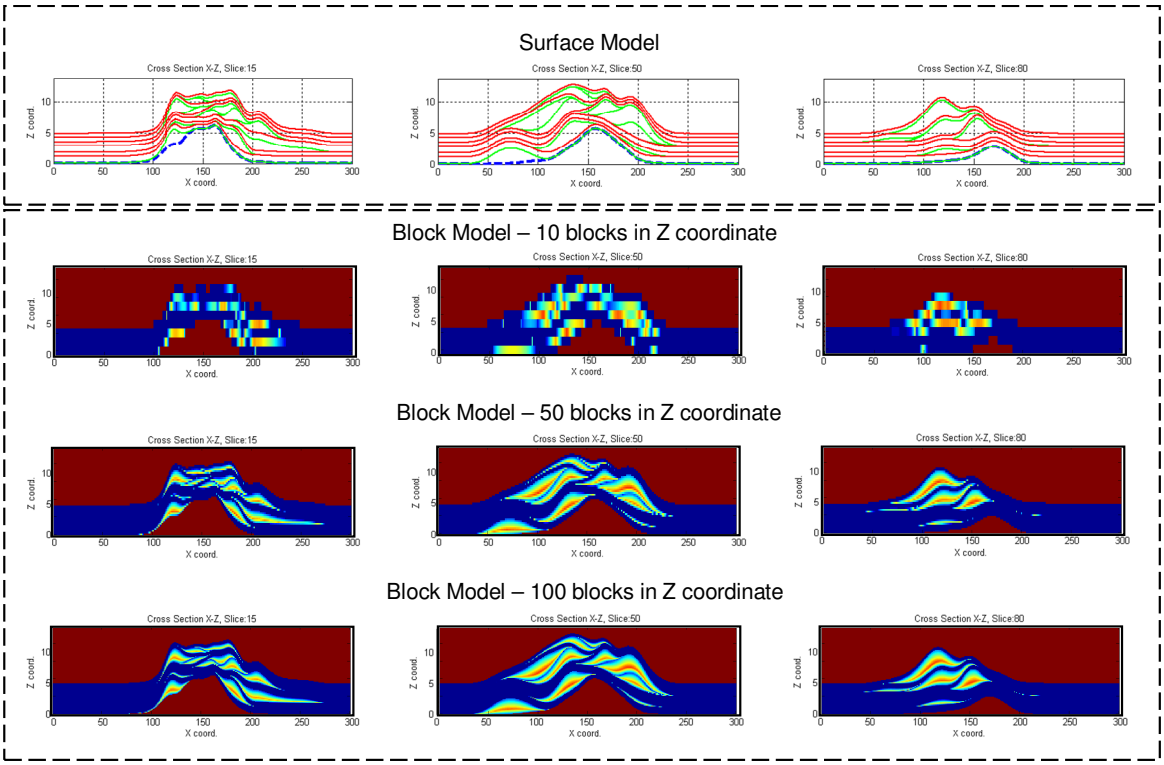


Figure 5-2: Discretization issues

5.2 Non-conditional Simulation

This section is meant to show the realism achieved by the model proposed in this report. Figures 5-3 to 5-7 show different views of a 12-lobe simulation realization.

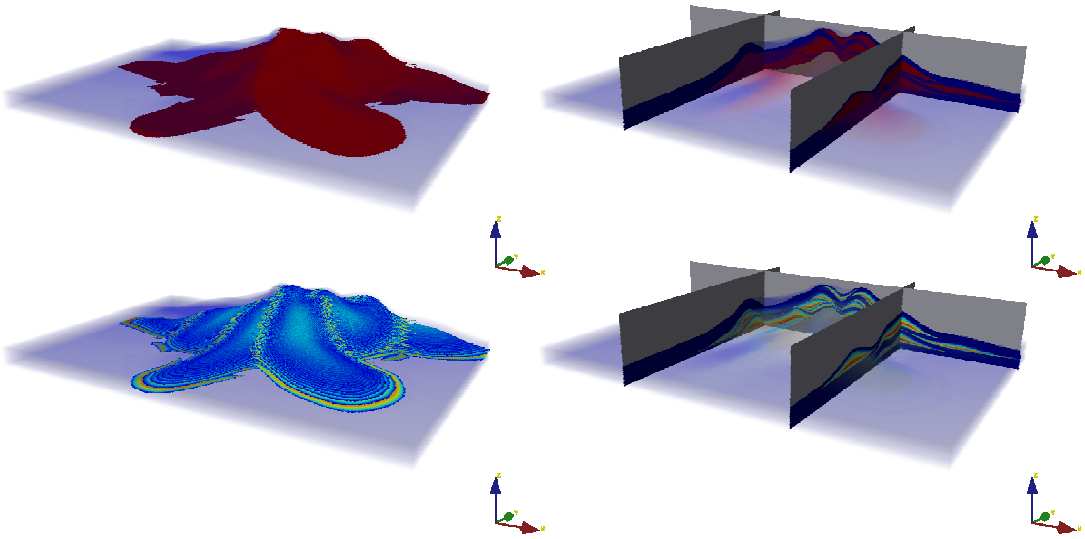


Figure 5-3: Isometric view of non-conditional simulation.

The realism of the realization is improved considerably with respect to classic object based simulation. In Figure 5.3 we can see that the geobodies point in the downhill direction, which is the main reason to incorporate drainage area calculation in the model. Also the geobody stacking is improved since they tend to be deposited in the low zones following the flow directions, which reflects consistency between base topography and flow events simulated.

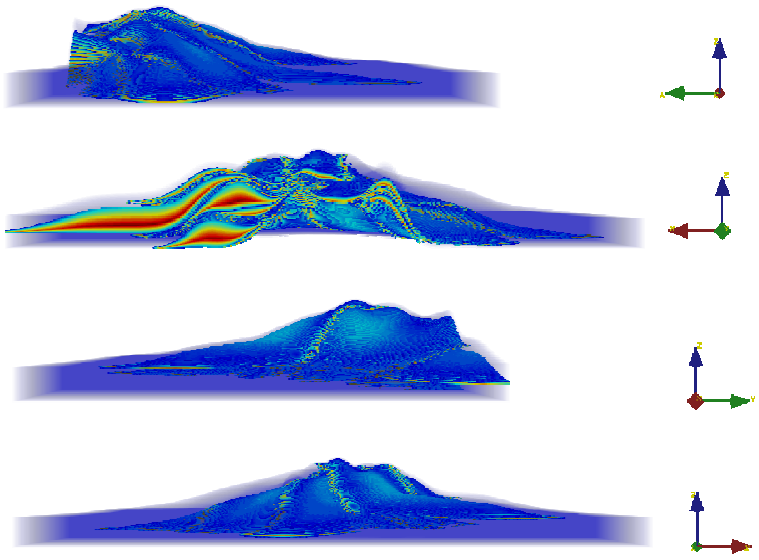


Figure 5-4: Cross section views of property distribution for a non-conditional simulation.

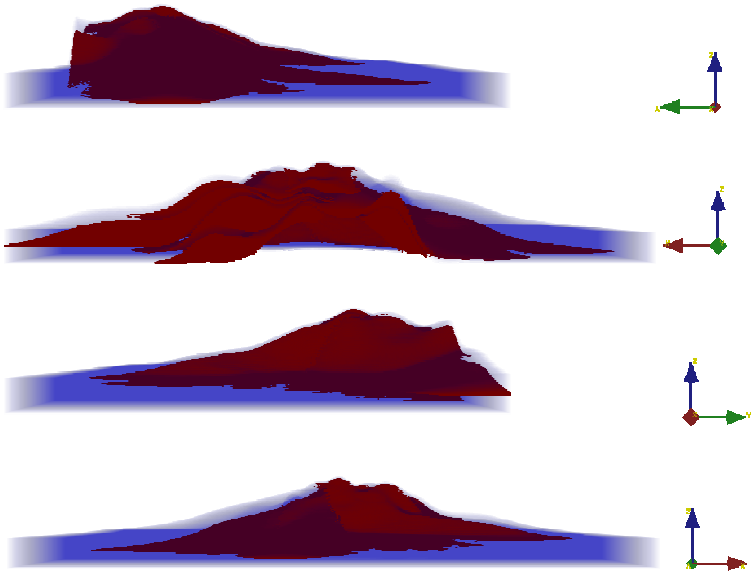


Figure 5-5: Cross section views of facies model for a non-conditional simulation

The isometric view in Figure 5-6 and the top view in Figure 5-7 show clearly that the orientation of the lobes follow the flow path and the stacking is related to the topography.

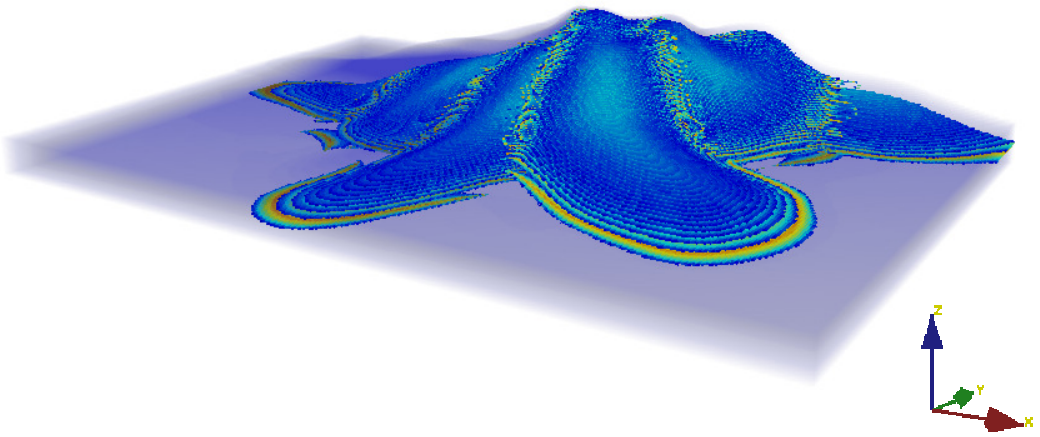


Figure 5-6: Isometric view of non-conditional simulation.

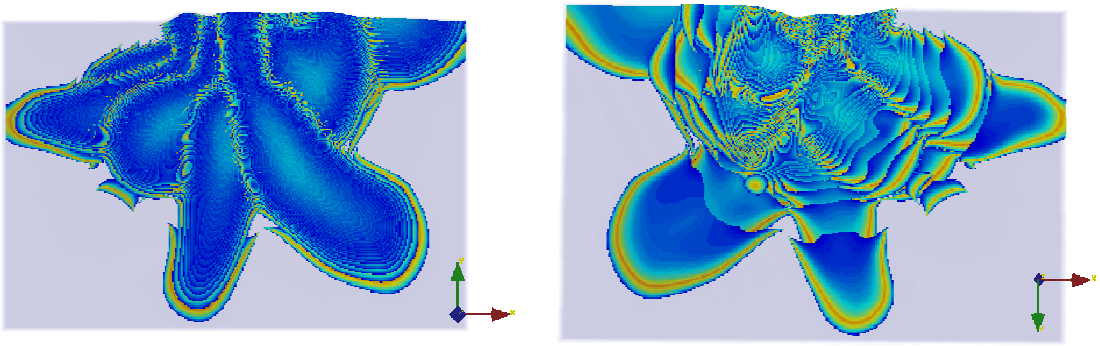


Figure 5-7: Top and bottom view of non-conditional simulation.
Secondary variable computed using property distribution 3 explained in chapter 4 is displayed.

The parameters used in the model simulation and the statistics obtained from the grid-based result are presented in Table 5-1. Notice that the realization is $300x200x100$ gridblocks, and among them only 1.726.617 blocks are simulated as either sand or shale (28.7% of the block). The remaining blocks are located below the initial topography or above the last layer formed by the stacked lobes and shale layers.

Table 5-1: Statistic of non-conditional simulation.

Sand percentage	30.62%
Sand Blocks	528649
Shale percentage	69.38%
Shale Blocks	1197968
Sand/Shale ratio	0.44
Dx	2
Nx	300
Dy	2
Ny	200
Dz	1
Nz	100
Prc. Blocks >0.9	1.47%
Prc. Blocks >0.5	48.10%
Prc. Blocks >0.1	97.41%
Number of Lobes	0.12
Number Ind. Geobodies	4
IFGU threshold	12000 years
IFGU dep. Rate	5.00E-05

Table 5-1 also introduces the concept of connected or non-connected (independent) geobodies. There are multiple ways to define if a geobody is connect to another or not. In this case, given the correspondence with the way traditional reservoir simulators consider interaction between blocks, a six-neighbor scheme is taken. This is, two sand blocks belong to the same geobody if they are connected in a 6 neighbor scheme for regular grids. For the particular simulation shown in this section, although 12 geobodies were simulated, there are only 4 independent geobodies.

With respect to the speed of running the model, the results are very encouraging. Figure 5-8 shows the model performance for 25 independent eight-lobe simulations (using parameters in table 5-1). Most of the simulations took between 50 to 100 seconds. There is only one particular simulation that took longer than 100 seconds[1]. This outlier can be explained by the condition imposed on the length and orientation of the influence area. Sometimes, given the changes in topography that happen throughout the simulation, obtaining an influence area that meets the conditions becomes quite hard. This

[1] On a Intel single core 2,659 MHz machine.

phenomenon is very dependent on the initial topography. Highly steeped topography should get stuck less often in this kind of situation. Given the irregular initial topography used in this example, it is normal for the model to get stuck finding an appropriate anchor point after a series of successfully modeled flow events.

The fluctuations in the time taken for each particular simulation, even though the same parameters were used for all of them, are related to the stochasticity of the model. Sometimes the simulation goes very smoothly, but it is not guaranteed since irregular intermediate topographies, that make the constraints more complex, can be generated during the course of the simulation.

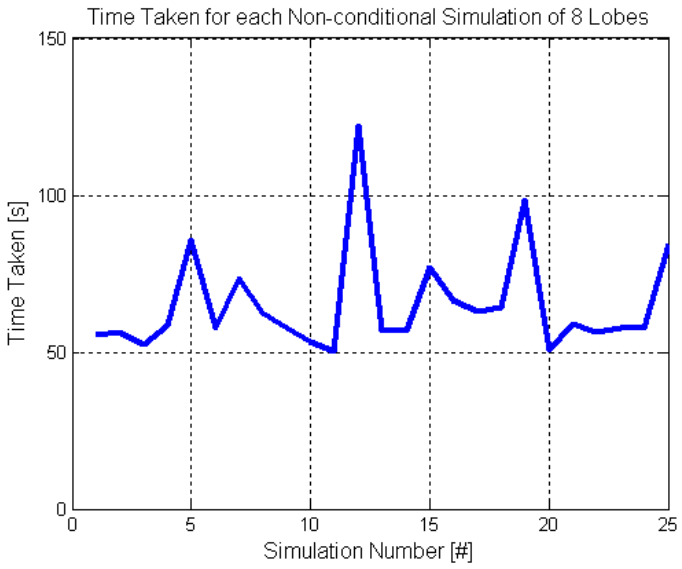


Figure 5-8: Time taken for each non-conditional simulation of 8 lobes.

5.3 Conditional Simulation

In this section eight-lobe simulations are conditioned to well data (Figure 5-9) using the techniques explained in chapter 4. The data conditioned are 3 wells, specifically the sand intervals in them. With respect to the intervals above and below the sand formations, it is

only required the realizations to match the facies type, but no particular condition with respect to length of those intervals is imposed.

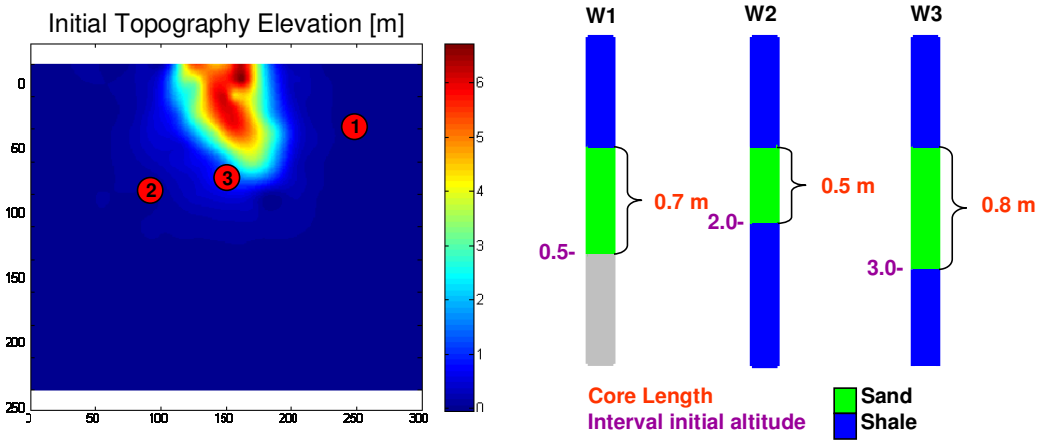


Figure 5-9: Wells conditioning the model.

Well #1 (Figure 5-9) shows a transition sand-shale, with the sand formation deposited immediately above the initial surface (represented in gray). Wells 2 and 3 show shales below and above the sand formation. Two conditional simulation outputs are shown in Figure 5-10. Although they look very different, the two of them match exactly the facies and thicknesses at the data locations. Figure 5-11 shows the perfect matched of well data achieved by the simulation #1.

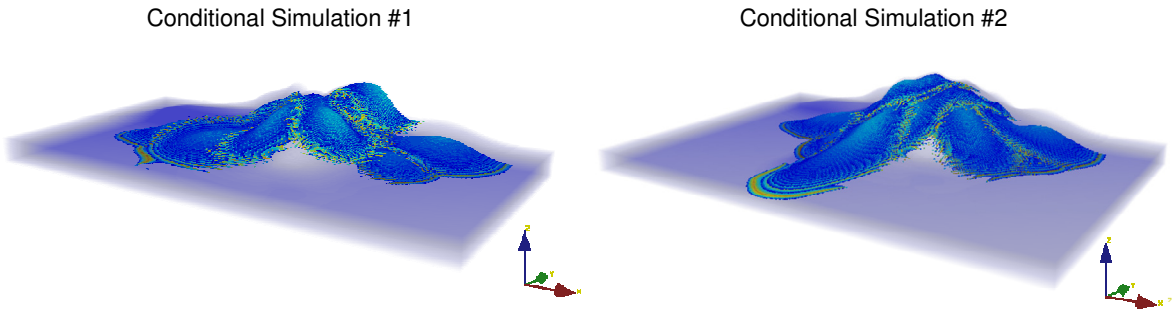


Figure 5-10: Conditional simulation outputs.

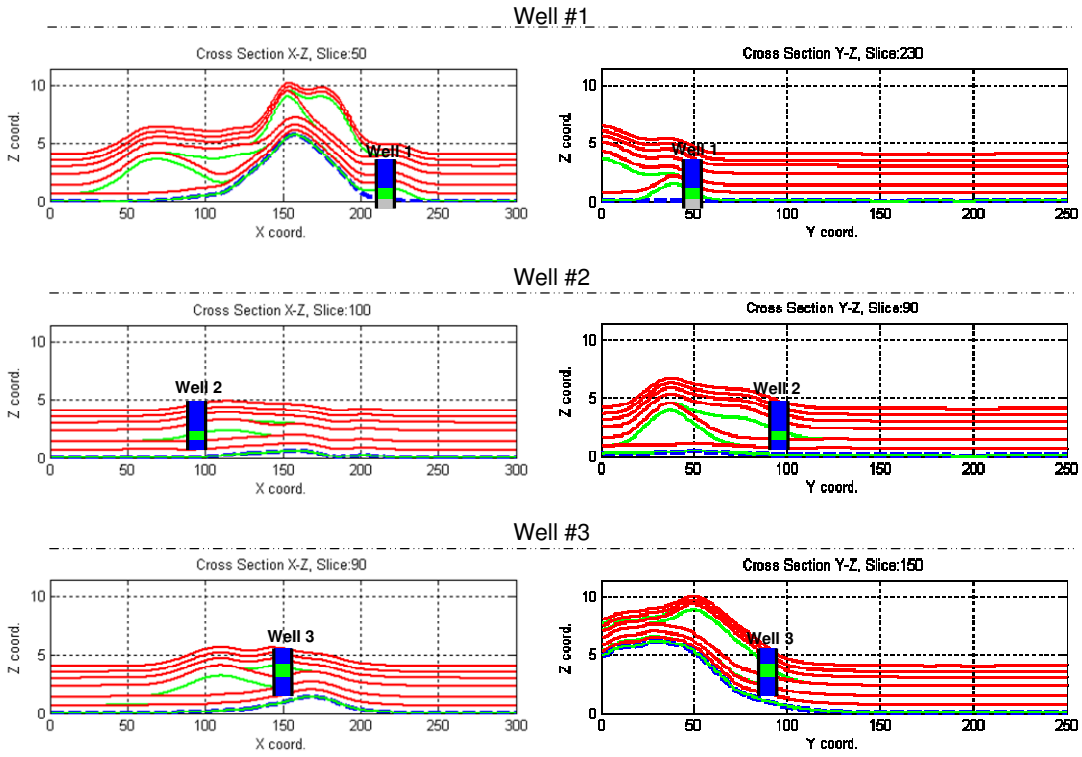


Figure 5-11: Hard data matching – Conditional Simulation #1.

Table 5-2: Conditional Simulations Statistics Comparison.

	Conditional Sim. 1	Conditional Sim. 2
Sand percentage	14.29%	22.29%
Sand Blocks	249867	339097
Shale percentage	85.71%	77.71%
Shale Blocks	1498262	1181895
Sand/Shale ratio	0.17	0.29
Dx	2	2
Nx	300	300
Dy	2	2
Ny	200	200
Dz	1	1
Nz	100	100
Prc. Blocks >0.9	1.64%	1.42%
Prc. Blocks >0.5	48.50%	47.78%
Prc. Blocks >0.1	97.47%	97.30%
Number of Lobes	12	12
Number Ind. Geobodies	3	4
IFGU threshold	12000 years	12000 years
IFGU dep. Rate	5.00E-05	5.00E-05

Table 5-2 shows a comparison between the two conditional simulations statistics. Although both model match successfully the data, their statistics look very different (the model sizes are the same). The major differences are found in the sand/shale percentage and the number of independent geobodies simulated. These two features are very relevant in flow simulation. Therefore, even though the models are consistent with the data, special attention has to be paid on the model statistics. Section 5.4 goes through the parameters than can be manipulated in order to have a better control on these simulation features.

The time taken for each conditional simulation on average is slightly above the time taken for non-conditional simulations with the same set of initial parameters. Although the number of conditioning wells is small, the performance shown by this approach is very good. Considering that very little time is required for generating a conditional realization, the goal of obtaining a complete set of conditional realization in fraction of the time taken by a purely object-based or process-based approach is achieved. This goal should be potentially achieved even in the case that more wells are conditioning the model, although more work has to be done regarding this.

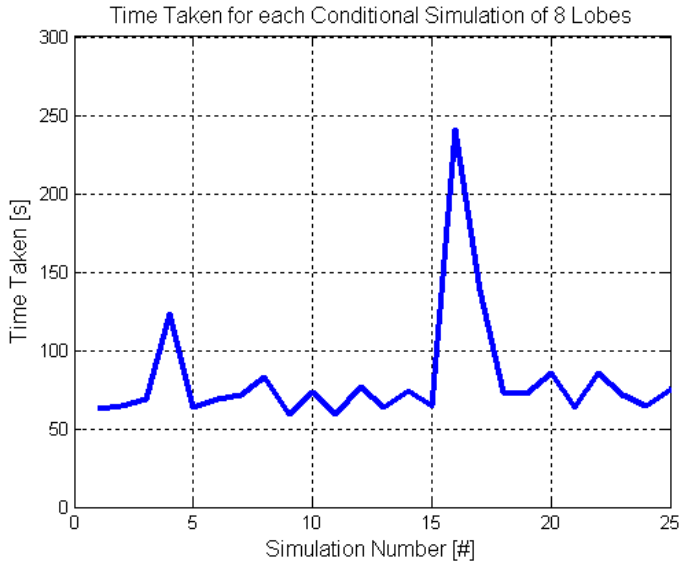


Figure 5-12: Time taken for each conditional simulation of 8 lobes.

Some modifications are required in the implementation to make the model able to account for multiple data, but once implemented the simulation time should not increase drastically.

There are two events in which the time goes over 100 seconds, and once it goes even above 200 seconds. The explanation to the situation are the possible difficulties to find the conditional simulation area, since once it is found the model, due to implementation and conditioning to hard data techniques used, runs smoothly.

5.4 Statistics

The statistics obtained from the process-based simulation and used as input data in the model are extremely important since they will determine important features such as facies proportion and geometry of geobodies. This section studies the importance of the most relevant parameters extracted from the process-based model and gives clues on how this parameter can be used in to order to obtain simulations that meet the user requirements.

5.4.1 Fine-grained Unit threshold

Fine-grained units are deposited after a flow event only if the time (in years) that passes before the following flow event (drawn from a CDF) is larger than a certain threshold. If the time is above this threshold, the thickness of the shale layer is given by this time multiplied by the depositional rate in meters per year. Using the depositional time CDF between two flow events shown in Figure 5-13 and a depositional rate of $4 \cdot 10^{-5}$ [m/year], we study the importance of the threshold, in terms of amount of each facies in the realizations.

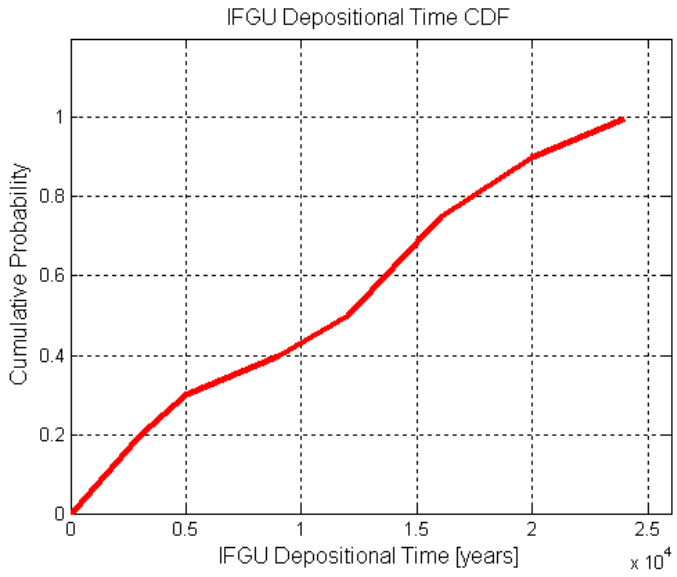


Figure 5-13: IFGU depositional time CDF.

Figure 5-14 shows the number of blocks of sand and shale for an eight-lobe non-conditional simulation as a function of the depositional rate threshold. The figure shows a very close to constant number of sand blocks as the threshold changes. This is expected since the threshold value is not a parameter involved in the geobody geometry. On the other hand, the number of shale blocks curve is highly affected by this threshold. It shows a clearly inverse correlation, reaching a point at which no shale layers are present. This is because values above the threshold were never drawn from the CDF.

After this analysis it is possible to say that the depositional time threshold is a good controller of the amount of shale, which can be used in case a certain sand/shale proportion is required.

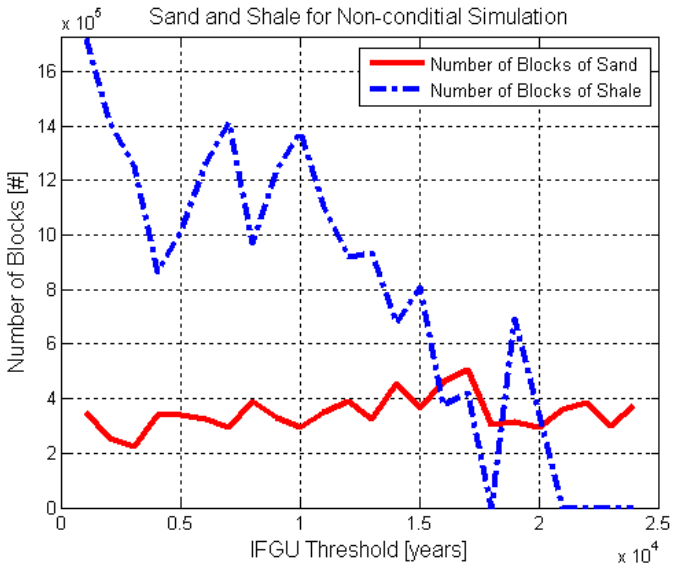


Figure 5-14: Sand and shale blocks v/s IFGU threshold.

5.4.2 Fine-grained Unit CDF

After studying the importance of the depositional time threshold for the intermediate fine-grained unit, we kept this parameter constant in order to assess the importance of the CDF shape in the sand/shale proportion shown by the model realizations. The depositional rate is set at $4 \cdot 10^{-5}$ [m/year], whereas the threshold is 12,000 [years]. That is, there will be an IFGU only if the time between flow events is equal or larger than 12,000 [years].

In this test, the end-point of the time that passes between depositional events CDF is changed. Linear CDFs stating from (0,0) are assumed. For each end-point assessed, 6 realizations were simulated. Figure 5-12 should the number of blocks for each facies for a given CDF end-point. Sands are not affected for changes in this parameter, which is natural since this particular CDF is not related to the geobodies shapes. On the other hand, the number of shale blocks increases directly with the end-point of the cumulative distribution. This is due the fact that a larger CDF end-point brings larger intervals between flow events, which makes the condition of being above the depositional time threshold less restrictive and the thicknesses of the shale layers, given by depositional rate times the time between depositional events, thicker.

Changing the CDF seems to be a better way of controlling the amount of shale in the model realizations than changing the fine-grained unit threshold, since the variations in end point and number of blocks are better correlated, although the highly stochastic component is still clearly present.

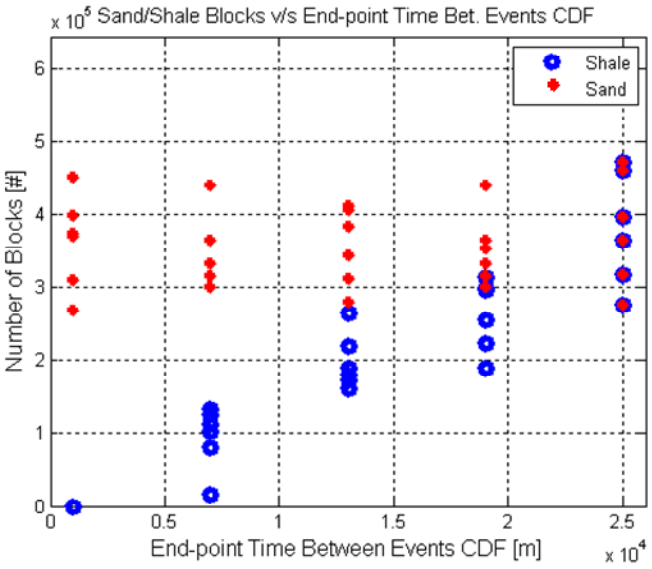


Figure 5-15: Sand and shale blocks v/s IFGU threshold.

5.4.3 Fine-grained depositional rate

After studying the IFGU threshold and time between flow events CDF (both kept constant in this section) we focus on the depositional rate. For this part the CDF shown in Figure 5-13 is used and the IFGU threshold is set at 12,000 years. Linear CDFs stating from (0,0) area assumed, whereas the end-points go from 10^{-5} to $24 \cdot 10^{-5}$ [m/year] in 24 equally-spaced intervals. Since only the depositional rate is changed, and this is a parameter directly related to the IFGU deposition, only the amount of shale changes considerable with the depositional rate. In Figure 5-16, the curve showing the number of sand block (in red) is almost constant, and just small fluctuations related to the stochasticity of the geobody geometry parameter selection are observed. The shale curve shows a clear, but erratic at the same time, tendency to increase with the depositional rate.

Although the depositional rate is an immediate component of the layer thickness, its correlation is less linear than it might be expected. This erratic behavior can be explained by having a large IFGU threshold, which makes uncommon the presence of shale layers, hence, although there is a high depositional rate, many times there is no shale deposited.

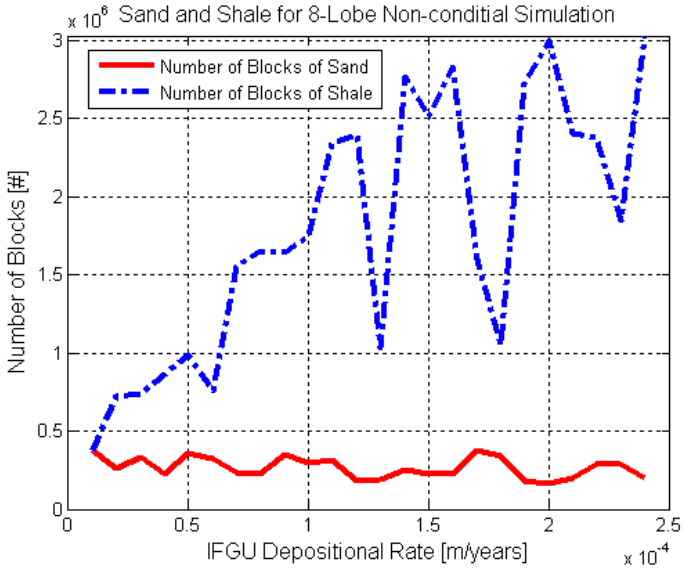


Figure 5-16: Sand and shale blocks v/s IFGU threshold.

An auto-control of facies proportion that combines perturbation on the depositional rate, IFGU CDFs, and IFGU depositional threshold can be implemented, but by doing this much of the stochasticity of the model would be lost since layer thicknesses would be assigned on the flight according to what is required, therefore decreasing the simulation levels of freedom.

5.4.4 Channel lobe length CDF

For studying the importance of the lobe length CDF, all the others parameters were kept constant. The depositional rate was set at 10^{-5} [m/years], the depositional threshold at 12,000 [years], and the depositional time between two event CDF used is shown in Figure 5-13. Linear lobe length CDFs stating from (0,0) is assumed. The end-points for the

CDFs go from 20 to 500 [m] in 4 equally-spaced intervals. For each CDF end-point, 6 non-conditional realizations consisting of eight lobes were simulated. Figure 5-17 presents the number of sand and shale blocks (separately) for these simulations with respect to the end-points in the lobe length CDFs. In the case of sand, the range in which the number of blocks is embedded increases with the CDF end-point. This certainly is correct, since a larger CDF end-point allows a wider range of geobody length, increasing the plausible amount of sand deposited (the number of geobodies simulated is kept fixed at eight). Notice that a too low lobe length distribution lower boundary would cause the model to run extremely slowly since according to the rejection rules used, the whole geobody (OB simulation) has to be inside the drainage area. Some type of tolerance can be applied to this constraint, but no matter what type of flexibility is taken, it should not harm the realism of the realization further than what might be obtained using influence and dependence area in the workflow.

With respect to the number of shale blocks in the simulation output, there is no correlation between them and the end-point of the lobe length CDF. This is expected, since the lobe length CDF affected directly the sand geobodies geometry, but has nothing to do with the shale layers.

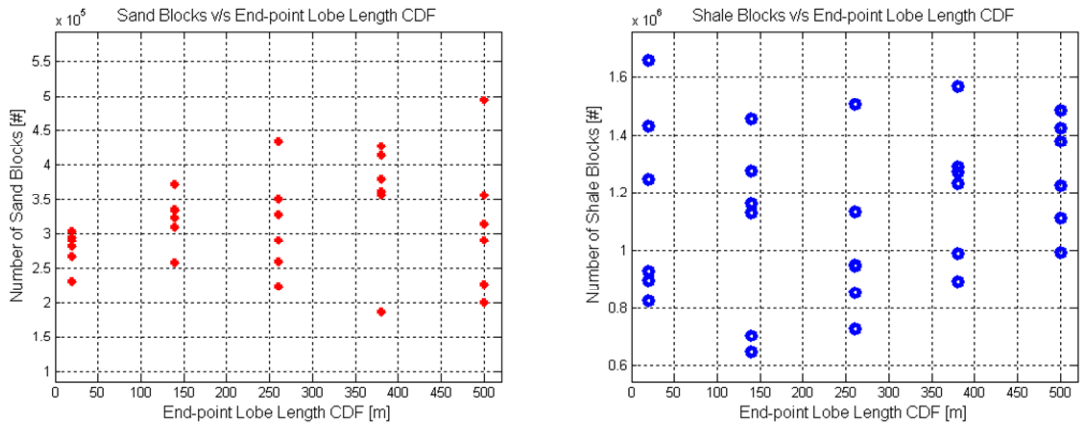


Figure 5-17: Sand and shale blocks v/s end-point lobe length CDF.

Although some degree of control can be achieved on the amount of sand in the model by changing the end-point of the lobe length CDFs, this is not enough to control the sand-shale proportion (Figure 5-18). A better study of the relationships between lobe length, lobe width, and channel width should be carried out in order to have a better control of the complete geobody geometry, which might lead to a better control over the facies proportions.

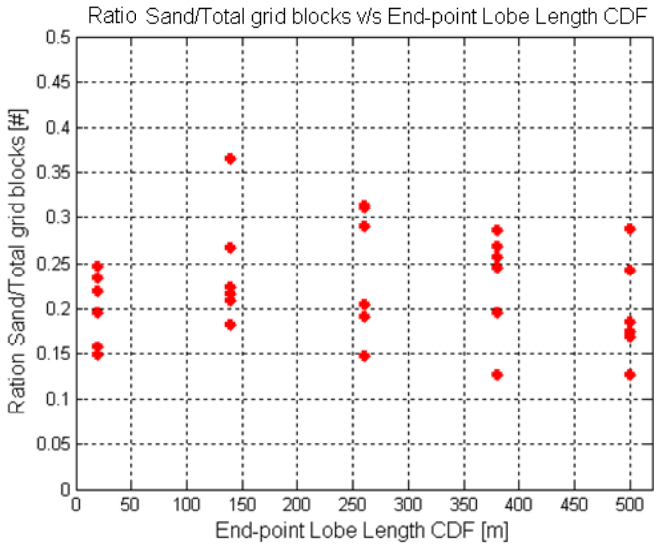


Figure 5-18: Sand-total blocks ratio v/s end-point lobe length CDF.

Chapter 6

6 Conclusions and Future Work

6.1 Summary and Conclusions

A workflow for constructing hybrid geostatistical models is proposed. Hybrid models break down the problem of simulating complex geological features into a set of smaller problems that can be faced using the most suitable approaches. The goals of hybrid models, in addition to simulating complex geologic features, are to speed up the simulation process, generate multiple and different realizations using a fixed set of initial parameters, and be possible to condition to hard data. The workflow starts by understanding the deposition environment and identifying the structures that have to be considered; these are the structures that play a role in the flow simulation. Then, the process continues with the parameterization of the important structures. These parameters must fully describe the geobody shapes and be able to be obtained from any related source of data, such as geologic studies, seismic interpretation, laboratory experimental data or process-based simulation. The realization generation follows a sequential process that starts with the simulation of major geobodies and continues with simulation of intermediate features. A two-dimensional realization is post-processed to achieve the final thickness map, which will be stacked on the base surface. In this part, geologic processes knowledge must be accounted for.

The initial deepwater turbidite hybrid model by Michael et al. (2008) is modified to account for the new aspects proposed in the methodology. The new model considers the changes that the previously simulated topography will bring to the simulation of the new lobe by using surface-based modeling (Influence Area and Drainage Area). Additionally, the algorithm generates realistic realizations, which are comprised of lobe-channel

structures, and intermediate fine-grained units. The proper simulation of the last is really important since its presence or absence might constitute a flow barrier.

The model conditioning is faced using two new approaches: Area expansion and grid deformation. Both show very good results in terms of computing time and geological consistency. Grid deformation has an enormous potential for conditioning simulations, but restrictions in its use must be applied in order to keep geologic realism.

Although the original model output is a layer-based realization, the assessment of the parameters is performed on a regular grid-based model. A very intuitive but efficient technique for extracting a block-based model from a layer-based model is presented. This technique generates block-based models in different resolutions from the same layer-based model.

Simulation input parameters such as depositional rate, CDF end-points, and depositional threshold can be used effectively as facies proportion controllers. Based on this, an on-the-fly control for facies proportion can be implemented, but special concern should be put on not losing the model stochasticity.

Once the facies reservoir model has been generated considering conceptual models and geological rules, it has to be populated with petrophysical properties. A secondary variable, which considers the relative location of a grid cell in the geobodies and also the relationship between different geobodies (given by the maximum lobe thickness allowed), is introduced. This secondary variable can be used efficiently as a trend input for any two-point geostatistics algorithm.

6.2 Recommendations for Future Work

There are still many improvements, extensions and tests that can be added to this work.
A few important possible future works are listed below:

- Studying anchor point decreasing trend relationship with the conditional simulation area. The decreasing trend used in the anchor point drawing process limits the plausible locations of the anchor point. This unfortunately, in cases when anchor is strictly needed far away from the sediment source (conditional simulation), increases considerably the simulation time.
- Dependence between geobody geometry-related parameters should be studied. Relationships such as flow event strength to lobe width and flow event strength to lobe length should be explored and incorporated to the model in order to make the simulation results more realistic from a geological point of view. Laboratory experimentation can become an extremely good source of data in this regard.
- Parameters such as fluid viscosity can be linked to geobodies geometry. Also, fluid properties and lobe deposition should be studied.
- An interesting topic of research can be the local dependency of property distribution trends to the underlying topography.
- Further study on the proper simulation of intermediate fined grain unit is really important, since its presence or absence might constitute a flow barrier.
- Finally, flow simulation should be run on model populated with petrophysical properties. These petrophysical properties should be generated accounting for trends. Conditional and non-conditional simulations can be compared to reference simulations.

References

Arpat, B. G., 2005, Sequential simulation with patters, Stanford Ph.D. Dissertation, pp. 68-70.

Deutsch, C. V. and Journel, A. G., 1998, GSLIB: geostatistical software library and user's guide, Second Edition, Oxford University Press, New York, 369 pp.

Deutsch, C. V., 2002, Geostatistical reservoir modeling, Oxford University Press, New York, 376 pp.

Deutsch, C. V., Y. Xie, and A. S. Cullick, 2001, Surface geometry and trend modeling for integration of stratigraphic data in reservoir models: 2001 Society of Petroleum Engineers Western Regional Meeting, March 26-30, Bakersfield, California, SPE paper 68817.

Fairfield, J., and Leymarie, P., 1991, Drainage network from grid digital elevation models, Water Resources Research 27 (5): 709-717.

Haldorsen, H. H., and D. W. Chang, 1986, Notes on stochastic shales; from outcrop to simulation model, in L. W. Lake and H. B. Carrol, eds., Reservoir characterization: London, Academic Press, p. 445-485.

Haldorsen, H. H., and L. W. Lake, 1984, A new approach to shale management in field-scale model: Society of Petroleum Engineers Journal, April, p. 447-457.

Lantuejoul C., 2002, Geostatistical simulation: Models and algorithms, Springer-Verlag, Berlin.

Mark, D. M., 1988, Network models in geomorphology, Chapter 4 in Modeling in Geomorphological Systems, Edited by M. G. Anderson, John Wiley., pp. 73-97.

Michael H., Li H., Li T., Boucher A., Gorelick S., and Caers J., 2008, Combining methods for geologically-realistic reservoir simulation, Geostat '08, VIII International Geostatistics Congress, 1-5 December, Santiago, Chile.

O'Callaghan J. F., and Mark, D. M., 1984, Extraction of drainage networks from digital elevation data. Comp. Vis. Graph. Image Proc. 28:323, p. 344.

Pyrcz, M. J., and C. V. Deutsch, 2003, Stochastic surface modeling in mud rich, fine-grained-turbidite lobes (abs.): AAPG Annual Meeting, May 11-14, Salt Lake City, Utah, Extended Abstract, www.searchdiscovery.com/documents/abstract/annual 2003/ extend/7796, 6p.

Pyrcz, M. J., 2003, Integration of geologic information into geostatistical models, Ph. D. Thesis, University of Alberta, Edmonton.

Pyrcz, M. J., Cateneanu O., and C. V. Deutsch, 2005, Stochastic surface-based modeling of turbidite lobes: AAPG Bulletin, V. 89, no. 2, February 2005, pp. 177-179.

Strebelle S., 2002, Conditional simulation if complex geological structures using multiple-point statistics, Mathematical Geology, V. 34, no. 1, pp. 1-21.

Toivanen, P. J., 1996, New geodesic distance transform for gray scale images, Patter Recognition Letters 17, pp. 437-450.

Torboton D., 1997, A new method for the determination of flow directions and upslope areas in grid digital elevation, Water Resources Research 33 (2): 309-319.

Xingquan K. Zhan, Michael J. Pyrcz, and Clayton V. Deutsch, 2008, Advances in stochastic surface modeling of deepwater deposition systems, Geostat '08, VIII International Geostatistics Congress, 1-5 December, Santiago, Chile.

Appendix A

A. Model Description

An explanation in words of the modeling code, what it does, and what functions are called.

Functions are in blue
Parameters are in red
Inputs (e.g. input files read into Matlab) are in green

Description of turbidite_system_conditional:

This is the main program that calls everything else.

Parameters that can be specified (reference parameter values are shown):

`ifgu_threshold = 12000;`
Set the IFGU threshold [years]. If the time between low event is longer than the threshold, the model simulates IFGU.
`int_dep_rate = 5e-5;`
Average depositional rate [m/year] for time between lobes.
`nlobe = 8;`
Number of lobes
`trend_map_range = 0.3;`
Range for the trend map. It can go from 0-0.5. it is delimited by the shorter side of the model.
`s_source_windows = 60;`
Total width of the sediment source windows. Which is located in the middle of the northernmost boundary of the model.
`max_lobe_thickness_allowed = 3;`
Maximun thickness allowed for any lobe in the simulation[m].

Flags that have to be specified:

`Show_plots;`
Show all the intermediate altitude maps generated during a depositional event.
`Keep_plots_after_iteration;`
Keep all the plots after each depositional event. Only works if `Show_plots` is true.

`consider_erosion; max_erosion_fraction = 0.2;`
Add erosion to the simulation. `max_erosion_fraction` is the maximum erosion with respect to the maximum thickness allowed.
`add_noise;`
Add noise to the erosion and elevation map. Only used in non-conditional case.
`add_noise_ifgu;`
Adds noise to the IFGU simulation. Only in non-conditional case.
`smooth_surface; lmv_smoothing = 11;`
Smooth out the surface before computing the Influence area using local moving window algorithm. `lmv_smoothing` is the size of the size of the window.
`conditional; n_wells = 3;`
Make the simulation conditional to hard data. Otherwise the simulation does not consider the data files. `n_wells` tells the model how many wells are to be considered.

Inputs:

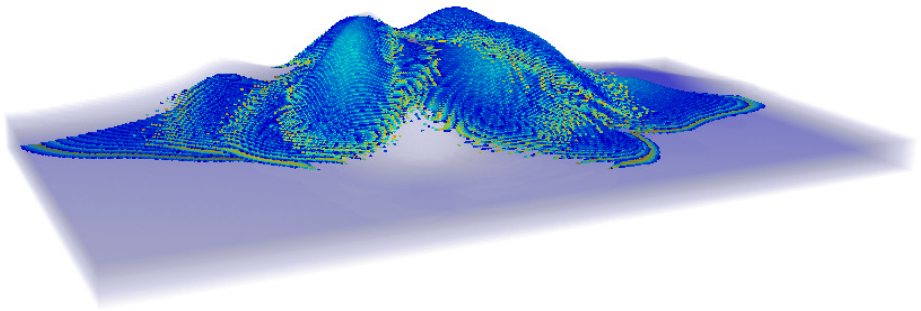
base_topo: The base topography. This is currently taken from the process-based model.
well_data folder: hard data from wells. Files located in the folder \code\well_data\
lobe_width_cdf: the CDF of the maximum lobe widths.
lobe_length_cdf: the CDF of the maximum lobe lengths.
max_lobe_thick_cdf: CDF of maximum lobe thickness.
channel_width_cdf: CDF of maximum channel width.
intermediate_depositional_time_cdf: years of depositional time for IFGU.
chan_detrend_totL_over_W_cdf: the CDF of the variations around the trend in the ratio of length to width of the channels.

Outputs:

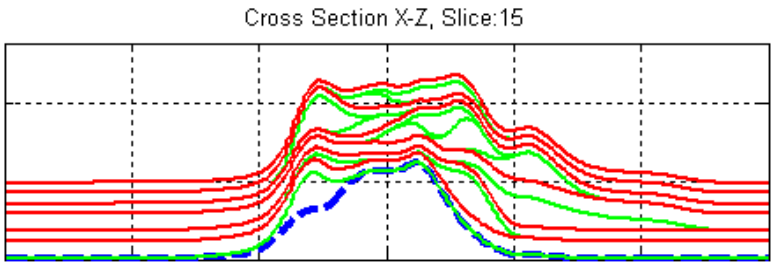
surface_cube: holds the top surface elevation of each layer, including the base_topo.
sim_thick_cube_strata: holds the simulated thickness if each layer.
sim_cat_cube_strata: holds the simulated category each layer.
sim_eros_cube_strata: holds the erosion thickness of each layer (negative)
tracklobe: tracks which of the layers are lobes and which are IFGUs. N-lobe for a simulated lobe layer, 0 for IFGUs, and -2 for non-simulated layers.

Visualization functions:

```
[model_facies model_sec_var model_facies_geoeas model_sec_var_geoeas] =
get_model(surface_cube,tracklobe,ny,nz,write_file,file_name,c,max_lobe_t
hickness_allowed,h);
```



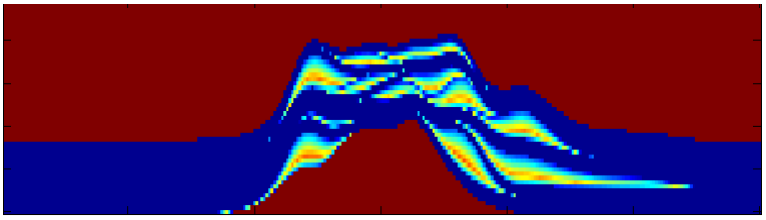
```
[log_facies log_sec_data zone_thickness]=
get_model_slice(surface_cube,tracklobe,nz,nslice,grid_block,plot_figure,
log_or_sec);
```




```
model_movie(surface_cube,tracklobe,n,code,delay,hold_image);
```

Shows a sequence of cross sections obtained using `get_model_slice`.

```
plot_model_slice(surface_cube,tracklobe,nslice,code,legend_flag);
```



Example on how to run a simulation and get the output file 'ex1.txt'

turbidite_system_conditional

```
[model_facies model_sec_var model_facies_geoeas model_sec_var_geoeas] =
get_model(surface_cube,tracklobe,150,100,true,'ex1.txt',0.3,3,0.3);
```

Functions called by turbidite_system_conditional:

- `get_dirname;`
- `read_cdf_data(lobe_length_cdf_data,verbose);`
- `trend_map_maker(g,center_point,trend_map_range);`
- `make_pfield_for_startpoint_tau2(g,base_topo,trend_map_starting_point,[]);`
- `read_harddata(conditional,M,N,n_wells,dirname);`
- `lmv_surface(E,lmv.smoothing,g);`
- `get_core_data(hard_data_location,dirname,g);`
- `get_simulation_area_conditional(E,[dataj datai],pfield_startpoint,g);`
- `get_simulation_area(E,pfield_startpoint,show.plots,g,25);`
- `get_core_data_in_SA(map,hard_data_location,dirname,g);`
- `cont(sim_area);`
- `get_angle(fliplr(start_point),contour_map,85);`
- `get_source_location(prev_sim_elev,[pos_j pos_i],s_source_windows);`
- `lobe_generator(start_point,s_source_location,angles,lobe_max_thick,g, verbose,Input_path);`
- `centerline_generator(s_source_location1,start_point1,g);`
- `find_nearest_point(d_chan,g,verbose);`
- `channel_shape_generator(across_dist,chan_detrend_totalL_to_width_cdf, lobe_max_thick,d_chan,g);`
- `lobe_thickness_assignment(closeBW,start_point,g);`
- `noisy_data('hard_data_deposition.txt',thickness_map,0.5,g)`
- `write_dimension_parfile_depo('sgsim_deposition.txt',g.nx,g.ny);`
- `lobe_cond_thickness_assignment(closeBW,start_point,dataj,datai,pos_data,length_core,max_lobe_thickness_allowed,g);`
- `curvature(flipud(prev_sim_elev));`
- `gradient8(flipud(prev_sim_elev));`
- `river_direction(I,start_point);`
- `erosion(sim_layer,profc,G,cond_thickness_map,min_dir,center_point,g,4/5);`
- `get_fgu(prev_sim_elev,fines_thick,0.5);`

Outline of workflow:

1.- Clear and close everything.
2.- Get the full file path.
3.- Simulation parameters and flags. Everything here is user defined.
4.- Load the surface. Might change if a different surface wants to be used.
5.- Make G-grid. Basically make a structure with the dimension of the model. It is easier to handle the dimensions by using this structure.
6.- Compute sediment source coordinates.
7.- Read in all input CDFs.
8.- Construct trend map.
9.- Initiate the matrices that will hold the model output.
10.- Assign first values to the previous matrices.
11.- Read hard data (only if conditional is true).

12.- Begin the simulation loop. Simulation continues until lobe_num <= nlobe.

- Smooth the surface

- Check if the lobe has to match the data necessarily. Case previous lobes did not go over the data location.

- Get the core data.

- Get conditional simulation area that allows the next lobe to match the data.

- Get non-conditional simulation area.

- Check if there is any data in this area.

- if not, the simulation continues as non-conditional. If there is data (`conditional_thickness = true`), the lobe has to meet thickness requirements.

13.- Generate lobe inside the simulation area.

- Get allowable angle range for lobe object-based simulation.

- Check is the thickness drawn from CDF is lower than the maximum thickness allowed.

- Iterate until generate a lobe in the simulation area. The tolerance can be added. No tolerance (ratio = 1), tolerance (1>ratio>0).

- Get the channel from the sediment source to the anchor point with channel width drawn from CDF.

- Perform morphological closing to avoid irregular boundaries in the lobe simulation (see Matlab help for more info).

- Check if the simulations is or not conditional.

- If not conditional, add noise in case it is required.

- If conditional, call lobe_conditional_thicknesss_assignment.

14.- Lobe thickness assignment. If not conditional noise can be added.

15.- Erosion calculation.

- Check if erosion max. value is lower than maximum fraction (e.g. 0.2) of max. lobe thickness allowed.

- Compute erosion components: Curvature, gradient (8 possible directions), and river-gradient direction relationship.

- Compute erosion map.

- Add noise in case the it is required and the simulation is non-conditional.

- Make sure that erosion does not go under the previous stacked layer.

- Save layers (`surface_cube`).

16.- Fine grained unit modeling.

- Compute fine grained unit thickness (average thickness for the unit).

- Check if depositional time drawn from CDF is larger than the threshold set by the user. In case it is, the IFGU simulation starts.

- Stack the layer just simulated (`surface_cube`) and assign flag (0) to `tracklobe` output.

17.- Re-start variables used in the loop. Simulation goes on until lobe_num > nlobe.