History Matching of Object-Based Stochastic Reservoir Models
Lin Y. Hu, Institut Français du Pétrole

Abstract
Object-based models can be regarded as arrangements of a population of geometrical objects in space. In reservoir modeling, these models are often used for describing meandering channel systems, fracture networks etc., where geological objects can be clearly identified. Unlike pixel-based models, object-based models can provide geologically realistic representations of reservoir heterogeneities even at the appraisal stage with few well data. Although an extensive literature on object-based models is available, little has been done for constraining these models to dynamic data (history matching). This is however of great importance for their application in reservoir engineering.

This paper first reviews the basic concepts of the widely used object-based Boolean model. Then, we present a methodology for calibrating Boolean simulations to dynamic production data. This methodology is based on a generalization of the gradual deformation method that was initially developed for calibrating pixel-based Gaussian-related reservoir models to dynamic data. Finally, a simple synthetic example is presented and the results show the validity of the above methodology. In particular, this methodology is potentially applicable to history matching of fractured reservoir models.

Introduction
In the last two decades, different stochastic models have been developed for describing reservoir heterogeneities of different depositional environments and at different scales. These models can be classified in three types: pixel-based models (e.g., Gaussian-related stochastic models), object-based models (e.g., Boolean models) and process-based models. Pixel-based models are relatively easy to be constrained by quantitative data but they are often unable to describe complex geological features, particularly at the field appraisal stage with few well data. On the contrary, process-based models can reproduce complex geological features but they are highly difficult to be constrained by quantitative data. In the case where geological objects can be clearly identified (fractures, faults, channels, vacuoles etc.), object-based models can be a good compromise between pixel-based and process-based models. There are many examples of geological modeling of fluvial-deltaic reservoirs using the object-based approach. For fractured reservoirs, Cacas et al. developed a specific object-based model to represent fracture swarms and sub-seismic faults. However, little has been done for constraining these models to dynamic production data. This is however of great importance for their application in reservoir engineering.

Constraining object-based models to production data requires the development of algorithms for the consistent deformation and migration of geometric/geological objects. In recent years, the gradual deformation method was developed for constraining pixel-based Gaussian-related stochastic models to production data. This method consists in iteratively optimizing combinations of independent realizations of a multi-Gaussian stochastic model until the constraints are satisfied. This method was successfully applied to several cases with data from synthetic and real oil fields.

In this paper, we first review the basic concepts of the most widely used object-based Boolean model. Then, based on a generalization of the gradual deformation method to non-Gaussian stochastic models, we develop algorithms for the gradual deformation of Boolean simulations, including non-stationary and conditional Boolean simulations. These algorithms are integrated in an inverse procedure for calibrating Boolean models to production data. Finally, a simple synthetic example is used to illustrate the applicability of this procedure.

Object-Based Reservoir Models
An object-based reservoir model can be regarded as an arrangement of a population of geological objects in the reservoir field. The spatial distribution of these objects is defined accounting for their interactions (attraction, repulsion, clustering etc.). A basic object-based model is the Boolean model that can be intuitively defined as the union of objects of identical natures. Object locations of the Boolean model are defined according to a Poisson point pattern with a constant intensity. Object shapes and sizes are independent from their locations. This model can be generalized by combining objects of different natures or/and by using a regionalized Poisson intensity. In addition, an algorithm was developed for conditioning Boolean simulations to well data. We review in
this section the basic concepts of the Boolean model and its conditional simulation algorithm.

Poisson point pattern. The Poisson point pattern (or process) characterizes the spatial distribution of objects of the Boolean model. A Poisson point pattern is a denumerable set of points distributed in the $m$ dimensional space $\mathbb{R}^m$. This point pattern has the following characteristics:

- Let $D$ be a field in $\mathbb{R}^m$. If the volume of $D$, denoted $|D|$, is finite, then the number of points in $D$, denoted $N(D)$, follows the Poisson distribution with mean $\lambda|D|$. That is

$$P[N(D) = n] = e^{-\lambda|D|} \frac{(\lambda|D|)^n}{n!} \quad \forall n \geq 0$$

The number $\lambda$ is the characteristic parameter of the Poisson point pattern. It gives the mean number of points to be found in a unit volume, and it is called the intensity or density of the point pattern.

- Let $D_1, D_2, ..., D_k$ be disjoint domains of $\mathbb{R}^m$, then the numbers of points in these domains denoted $N(D_1), N(D_2), ..., N(D_k)$ are mutually independent random variables.

- Conditionally to $N(D) = n_p$, these $n_p$ points in the field $D$ are independent and uniformly distributed in $D$. Hereafter, we only consider Poisson point patterns in a finite field $D$.

Boolean model. Let $A$ be a random objet (also called primary grain in literature). The construction of a Boolean model in the field $D$ consists of the following steps:

1. Generate a number $N$ that follows the Poisson distribution with mean $\lambda|D|$,  
2. Generate independently $N$ uniform points $X_i$ ($i=1,2,...,N$) in $D$ that constitute a Poisson point pattern of intensity $\lambda$,  
3. Generate independently $N$ objects $A_i$ ($i=1,2,...,N$) that have the same statistical characteristics as the random object $A$, and then  
4. Implant the $N$ objects in $D$ according to the Poisson point pattern.

The union of these objects, denoted $Z$ constitutes a random set called Boolean model:

$$Z = \bigcup_{i \in N} A_i(X_i)$$

The probability for an object $B$ being included in the complementary part of $Z$ is given by:

$$P(B \subset \bar{Z}) = e^{-\lambda|A \oplus B|}$$

where $|A \oplus B|$ means the volume of $A$ dilated par $B$. The Boolean model is completely characterized by formula (1) called its distribution function.

In particular, if the object $B$ is reduced to a point, one obtains the proportion of the volume occupied by $\bar{Z}$ in the field $D$:

$$q = e^{-\lambda|x|}$$

In most applications, the spatial distribution of objects is not uniform. The above stationary Boolean model with a constant intensity is then generalized to the non-stationary Boolean model by using a regionalized intensity function $\lambda(x)$.

Conditional simulation algorithm. The conditional simulation algorithm is a Markov iterative method. It allows to simulate a Boolean model in the field $D$ under the condition that two subsets $C_1$ and $C_0$ of $D$ belongs respectively to the union of objects $Z$ and to its complement $\bar{Z}$. The steps of this algorithm are as follows:

1. Implant a certain number $n$ of initial objects in the simulation field $D$ (dilated in order to avoid the edge effect), subject to respecting the conditioning to subsets $C_1$ and $C_0$.
2. Generate a random variable $I$ that is equal to $+1$ with probability $\lambda|D|/(\lambda|D| + n)$, or $-1$ with the complementary probability $n/(\lambda|D| + n)$.
3. If $I = +1$, generate a new object and implant it at a uniform location in $D$. If the addition of this object does not compromise the conditioning, add actually this new object to the simulation and add 1 to $n$, then return to step 2. Otherwise return directly to step 2.
4. If $I = -1$, select randomly one of the $n$ objects already implanted in the simulation field. If the removal of this object does not compromise the conditioning, remove actually the object from the simulation, subtract 1 from $n$ and return to step 2. Otherwise return directly to step 2.

The practical application of this algorithm requires to stop the iteration when the generated random set can be regarded approximately as a realization of the conditional Boolean model. A direct way to study the problem of convergence consists in evaluating the degree of approximation between the conditional distribution of the Boolean model and the experimental one obtained by simulation. Unfortunately, the numerical calculation of the conditional distribution is highly tedious. In practice, the following regional quantities can be evaluated:

- the percentage of the number of residual initial objects,
• the total number of the objects generated in the field of simulation,
• the distribution of the number of objects met by a point,
• experimental variograms etc.

The stabilization of these regional quantities indicates, to a certain extent, the global convergence of a simulation.

Note that certain initial objects, in particular those implanted at the transition zones between the conditioning sets \( C_1 \) and \( C_0 \), are often more difficult to be removed than the others. Moreover, the probability of selecting an initial object is very small when the total number of objects in the simulation becomes very large compared to the number of initial objects. In order to accelerate the disappearance of the initial objects, one can select first the initial objects instead of randomly selecting an object among all objects in the simulation field. The step 4 of the previous algorithm of simulation is then replaced by the two sub-steps:

4.1 if \( I = -1 \), select randomly one of the residual initial objects (if all initial objects are removed, go directly to sub-step 4.2). If the removal of this object does not compromise the conditioning, remove actually the object from the simulation, subtract 1 from \( n \) and return to step 2. Otherwise go directly to sub-step 4.2.

4.2 select randomly one of the \( n \) objects already implanted in the simulation field. If the removal of this object does not compromise the conditioning, remove actually the object from the simulation, subtract 1 from \( n \) and return to step 2.

Otherwise, return directly to step 2.

The addition of sub-step 4.1 within the algorithm of simulation makes it possible to first select an initial object when one tries to remove an object, thus to accelerate the disappearance of the initial objects. Note that when no more initial objects remain in the simulation field, or when the residual initial objects resist to be removed, the procedure of simulation overpasses sub-step 4.1 and we end up with the initial algorithm. This guarantees the convergence of the simulation with the modified algorithm towards the conditional Boolean model.

**History Matching of Object-Based Models**

History matching of an object-based model can be formulated as an optimization problem. The stochastic optimization consists in generating realizations of a stochastic model that reduce an objective function to a low-enough level. An effective optimization procedure requires the development of an algorithm for the consistent deformation of realizations of the stochastic model.

The gradual deformation applied to a Boolean simulation involves the smooth and consistent change of the size, the shape and the location of each object. Location coordinates and size/shape parameters are independent random variables. Locations of objects are characterized by Poisson point patterns. In this section, we discuss how to gradually migrate the stationary and non-stationary Poisson point patterns. This leads us to the algorithm for the local migration and in particular for the migration of objects of a conditional Boolean model. The gradual migration of a point pattern induces a smooth change of the spatial distribution of objects of the Boolean model. We also examine how to gradually add or remove points during the migration. In addition to the migration of the objects, the gradual deformation of a Boolean simulation involves also the change of the size and the shape of objects. This can be easily carried out by gradually changing the uniform numbers used for sampling of distributions of the size and the shape of the objects. All these algorithms for migration and deformation of objects are integrated in a coherent inverse procedure for calibrating Boolean models to dynamic production data.

**Objective function.** Let \( f^{obs} = (f_1^{obs}, f_2^{obs}, \ldots, f_p^{obs}) \) be the vector of production data observed in the reservoir field and \( f = (f_1, f_2, \ldots, f_p) \) the corresponding vector of the responses of the stochastic model \( Z \). For a given realization \( z \) of \( Z \), the values of \( f_1, f_2, \ldots, f_p \) are obtained through numerical fluid flow simulations. The stochastic optimization aims at generating realizations of \( Z \) that reduce the difference between the observations \( f^{obs} = (f_1^{obs}, f_2^{obs}, \ldots, f_p^{obs}) \) and the model responses \( f = (f_1, f_2, \ldots, f_p) \) to a low-enough level. We define the objective function as the sum of the weighted quadratic errors of the model responses with respect to the observations in the reservoir field:

\[
J = \frac{1}{2} \sum_{i=1}^{p} \omega_i (f_i - f_i^{obs})^2
\]

where \( \omega_i \) denotes the weight attributed to response \( f_i \). The \( f_i \) (\( i = 1, 2, \ldots, p \)) are functions of the vector \( z \), so is the objective function \( J \). The minimization of the objective function \( J \) implies at least two difficulties. First, the number of parameters (sizes, shapes and locations of all objects) is generally very large. Therefore, it is highly tedious to optimize the objective function directly with respect to the components of the vector \( z \). Moreover, in the frame of stochastic optimization, the realization \( z \) cannot be arbitrary: it must respect the spatial structure (e.g., the intensity function) of the stochastic model \( Z \). To overcome these difficulties, we resort to the gradual deformation method.

**Migration of stationary Poisson point patterns.** Consider first the migration of a realization of a stationary Poisson point pattern. Without loss of generality, we consider a Poisson point pattern in the \( m \)-dimensional field \([0,1]^m\). Let \( x_1 \) and \( x_2 \) be two independent points uniformly located in \([0,1]^m\). We define a trajectory between \( x_1 \) and \( x_2 \) as

\[
x(t) = G[G^{-1}(x_1) \cos t + G^{-1}(x_2) \sin t]
\]

where \( G \) stands for the standard Gaussian cumulative distribution function. It can be shown that for any \( t \), \( x(t) \) is a uniform point in \([0,1]^m\). When two points are fixed, the trajectory of the gradual migration between them is entirely determined. Changing the location of one of the two points...
will change the migration trajectory. Fig.2 shows several trajectories of migration from a set of uniform random points in $[0,1]^2$ to another set of uniform random points in $[0,1]^2$. We observe that all these trajectories are around the center point (0.5, 0.5). More generally, it can be shown that the trajectory defined by equation (2) is symmetric with respect to the center of the field $[0,1]^m$ for any dimension $m$. This suggests that even if the two points are isolated in a corner/side of the field, the trajectory of the gradual migration between them can still reach the opposite corner/side of the field.

Fig.3 shows a chain of 5 realizations of a Boolean model with elliptic objects. In this example, objects are gradually migrated while their shapes and sizes remain unchanged. Due to the relatively small variation step in the deformation parameter ($\Delta t = 0.01\pi$), two consecutive realizations of the chain are very similar. However, several steps of migration can significantly modify a realization.

**Migration of general Poisson point patterns.** Consider now the migration of a realization of a non-stationary Poisson point pattern. Objects of a non-stationary Boolean simulation are not uniformly located in space. Let $\lambda(x)$ be the intensity function of a general Poisson point pattern in field $D$. The number of points in $D$ is a Poisson random variable with mean

$$\lambda(D) = \int_D \lambda(x) dx.$$  

These points are distributed independently in $D$ according to the probability density function

$$f(x) = \frac{\lambda(x)}{\lambda(D)}, \quad x \in D.$$  

The simulation of the Poisson point pattern of intensity $\lambda(x)$ in $D$ can be performed in two steps:

- generate first a number $N$ according to the Poisson distribution of parameter $\lambda(D)$, then
- generate $N$ independent points in $D$ according to the probability density function $f(x)$.

Fig.4 shows an example of the intensity function $\lambda(x)$ and Fig.5 a realization of the Poisson point pattern of intensity $\lambda(x)$. If one utilizes the inverse distribution method to simulate the distribution $f(x)$, then the point $x$ corresponds to a uniform vector. Consequently, one can apply the gradual deformation method to migrate the point $x$ of the Poisson point pattern of intensity $\lambda(x)$. Fig.6 shows two examples of migration trajectories.

**Migration of objects of conditional Boolean simulations.** The above method can be straightforwardly extended to the migration in a local domain $S$ of any shape. As a matter of fact, the migration of a point in $S$ can be performed in $D$ by using the truncated probability intensity function:

$$f_S(x) = \frac{\lambda(x)1_{x \in S}}{\lambda(S)}, \quad x \in D, \quad S \subset D.$$  

Local migration is of great interest for deforming conditional Boolean simulations. By using the previous conditional simulation algorithm, one can simulate the Boolean model in $D$, subject to the conditions that two subsets $C_1$ and $C_0$ of $D$ belong to the union of objects and its complement respectively. Starting with a Boolean simulation and without compromising the conditioning to $C_1$ and $C_0$, the migration of the objects must be restricted to their respective domains defined according to their shapes and the conditioning subsets $C_1$ and $C_0$. For instance, consider an object $A$ including a subset $C_{1,d}$ of $C_1$ and excluding $C_0$. We admit that during the migration of the object $A$, it must always include $C_{1,d}$ and excluding $C_0$. Thus, the authorized domain of migration of this object is:

$$D_A = \{x : C_0 \cap A(x) = \emptyset; C_{1,d} \subset A(x)\}.$$  

Fig.7 shows an example of three conditioning points and the eight possible domains of migration of an object (a disc). For instance, if the disc must cover the points (a) and (b) but avoid the point (c), then its center is allowed to migrate in domain 3.

**Migration while gradually removing or adding points.** The above migration algorithm maintains the number of points of the initial Poisson point pattern. To assure that the number of points in $D$ of a Poisson point pattern follows the Poisson distribution of parameter $\lambda(D)$, it is necessary to be able to migrate a realization of $N_1$ points (pattern 1) to a realization of $N_2$ points (pattern 2). This implies either the removal or the addition of points during the migration. The number of points to be removed or to be added is determined by the chain of Poisson numbers between $N_1$ and $N_2$. The algorithm of migration is as follows:

1. construct a chain $\{t\}$ of Poisson numbers between $N_1$ and $N_2$ using the gradual deformation algorithm;
2. compute the maximal number $N_{\max}$ of the chain $\{t\}$;
3. complete the pattern 1 of $N_{\max} - N_1$ points and pattern 2 of $N_{\max} - N_2$ points;
4. for each pattern, classify the points from 1 to $N_{\max}$ (all initial points are classified at the beginning);
5. compute the trajectory of migration from the point $n$ of the pattern 1 to the point $n$ of the pattern 2 ($n = 1, 2, \ldots, N_{\max}$);
6. at each state $t$ of the realization chain, remove the last $N_{\text{max}} - N(t)$ points.

**Gradual deformation of the intensity function.** In addition to the migration and deformation of the objects, the gradual deformation of a Boolean simulation may also involve the modification of its intensity function. This is because in practice, the intensity of objects is often uncertain and must also be regarded as a fitting (calibration) parameter. The gradual deformation of the continuous intensity function can be performed similarly to that of a continuous Gaussian random field.

For a Boolean model with a large number of small objects (Fig.8), the Poisson intensity dominates the hydrodynamic behavior of the model. On the other hand, for a Boolean model with a small number of large objects (Fig.9), the hydrodynamic behavior of the model can be sensitive to all parameters: location, size, shape and intensity of objects.

**Iterative calibration procedure.** Starting with an initial realization $z_0$ of $Z$ and another realization $z_1$ of $Z$ independent of $z_0$, we build a chain of realizations $z_t(t)$ by using the above algorithms of parameterization. Then we minimize the objective function with respect to parameter $t$, and resulting in an optimized realization $z_t(t_{\text{opt}})$ that improves or at least maintains the calibration to production data. In this way, the high dimensional optimization problem is reduced to a one-dimensional one, and moreover the optimized realization $z_1(t_{\text{opt}})$ is actually a realization of $Z$.

However, the above optimization might not reduce the objective function to a low-enough level. It is then necessary to repeat the same procedure with a new realization chain $\tilde{z}_t(t)$ built by combining $z_1(t_{\text{opt}})$ and another independent realization $\tilde{z}_2$ of $Z$. This procedure is iterated until a satisfactory calibration is reached.

Unlike in the case of continuous Gaussian-related models, the objective function of a realization chain of an object-based model is in general not differentiable with respect to the deformation parameter $t$. This is due to the discrete nature of object-based models. Consequently, gradient based optimization algorithms cannot be applied and we must turn to algorithms for optimizing non-differentiable functions. For instance, the golden section search method\textsuperscript{12} can be used for the above one-dimensional optimization problem. When more parameters are used, we can resort to simplex method.\textsuperscript{12}

**Example: Calibration of Boolean Model to Water-Cut Data**

A simple synthetic example is intended to illustrate the above methodology for calibrating Boolean simulations to dynamic production data.

**Reference case.** Consider a simple two-well case with a water injector located in the lower-left corner and a producer in the upper-right corner of the reservoir field (Fig.10). The dimension of the field is 1500m X 1500m. Using Boolean simulation, we build a reference model composed of two lithofacies: the union of twenty identical objects of radius equal to 150m and its complement (Fig.10). The objects are randomly implanted in the field while avoiding covering the two wells. The reservoir model is discretized into 100 X 100 blocks for fluid flow simulation. The objects are flow barriers with permeability equal to 50md and porosity equal to 5%. The permeability and the porosity of the complementary lithofacies are respectively 1000md and 20%. The injected water, displacing oil and flowing through the heterogeneous medium, is then partly recovered by the producer. The fractional water flow (water-cut) of the producer is closely related to the spatial distribution of the objects. Fig.11 shows the curve of water-cut of the reference model. The objective of the study is to generate realizations of the same Boolean model and to calibrate them to the curve of water-cut.

**Behavior of a realization chain.** From two other independent realizations (Fig.12) of the above Boolean model, we build a chain of realizations. Note first that the curves of water-cut of these two realizations are quite different between them and also different from that of the reference realization (Fig.13).

The objective function is defined as the sum of the square differences between the values of water-cut at each time step of the reference realization and those of each realization of the chain. Fig.14 shows the evolution of this objective function versus the sequence number of the realizations of the chain. Realization 1 corresponds to that of Fig.12(a) and realization 101 corresponds to that of Fig.12(b).

The optimal realization of this chain is the realization 98 (Fig.15) which corresponds to the deformation parameter $t = 0.485\pi$. This realization is visually not very different from realization 101 (Fig.12(b)) corresponding to $t = 0.5\pi$. However, the first one allows a good calibration to the curve of water-cut contrarily to the second one (Fig.16). This shows that the hydrodynamic behavior of a Boolean simulation is very sensitive to the geometrical distribution of its objects.

The irregularity of the objective function of a realization chain is typical of a lithofacial reservoir model. Let us examine more in detail the brutal change of the objective function from realizations 130 to realization 131 (Fig.17). Their curves of water-cut are quite distinct: the water breakthrough time in the producer of realization 131 is much later than that of realization 130 (Fig.18). This is because, from realization 130 to realization 131, small displacements of some objects destroy a flow path between the two wells. Fig.19 show the maps of water saturation, after 10 years of injection and production, of realizations 130 and 131 and the reference realization. These maps confirm the above observation.

**Iterative calibration.** Starting respectively from the two independent realizations of Fig.12, we applied the procedure of iterative calibration to the curve of water-cut. This calibration involves only the migration of objects. The golden section search method is used for the optimization of each realization chain. From the initial realization of Fig.12(a), we obtain a calibrated realization after 11 iterations (Fig.20, 21...
and 22). From the initial realization of Fig.12(a), we obtain a calibrated realization after only 4 iterations (Fig.23, 24 and 25).

Conclusions and perspective
A consistent methodology is presented for history matching of the object-based Boolean model. In this methodology, the parameterization of the Boolean model is based on the generalized gradual deformation method. We developed an algorithm for the gradual migration of general Poisson point patterns while preserving their intensity functions. This algorithm was extended to the local migration and to the migration of objects of conditional Boolean simulations. We also proposed an algorithm for gradually removing or adding points/objects. In addition, the size and the shape of objects can also be gradually deformed for history matching. Furthermore, when the Poisson intensity function is uncertain, it can also be gradually deformed.

Although the gradual deformation of a Boolean simulation results in smooth change of the model geometry, the hydrodynamic behavior of a realization chain may still have discontinuity due to the discrete nature object-based models. This requires implementing algorithms for optimizing non-differentiable objective functions. For instance, the golden section search method can be used for simple one-dimensional optimization problem. In real applications, several deformation parameters are often necessary and the simplex method can be used.

Preliminary results from a simple synthetic example show the validity of the above methodology. Further work will be its validation on Boolean simulations with more complex objects and regionalized intensity functions. Our ongoing research is also focussing on the extension of the above methodology to history matching of fractured reservoir models.

References
Fig. 1 – Example of an object-based model representing faults and fracture swarms in a reservoir field.

Fig. 2 – Trajectories of migration from a set of uniform random points (dots) in $[0,1]^2$ to another set of uniform random points (rectangles) in $[0,1]^2$.

Fig. 3 – A chain of 5 realizations of a Boolean model with elliptic objects (in white color) and with the step of the deformation parameter $\Delta t = 0.01\pi$. Objects are gradually migrated without changing their shapes and sizes.
Fig. 4 - Intensity function of a general Poisson point pattern

Fig. 5 - Realization of the general Poisson point pattern with the intensity function shown in Fig. 4.

Fig. 6 - Two examples of trajectories of migration between the two points (large block dots) of the general Poisson point pattern with intensity function shown in Fig. 4.

Fig. 7 - Location of three conditioning points and eight possible domains of migration of a disc.
Fig. 8 - Non-stationary Boolean simulation with a large number of small objects.

Fig. 9 - Non-stationary Boolean simulation with a small number of large objects.

Fig. 10 - Reference realization generated by Boolean simulation.

Fig. 11 - Water-cut versus time (day) at the production well resulting from a fluid flow simulation on the reference realization.

Fig. 12 - Two independent realizations conditioned to the wells.
Fig. 13 - Comparison between the curve of water-cut of the reference realization and those of the two independent realizations in Fig. 12(a) and (b).

Fig. 14 - Objective function of a chain of 201 realizations when the parameter of gradual deformation \( t \) varies between \( 0 \) and \( \pi \).

Fig. 15 - Realization 98 of the realization chain of the Boolean model.

Fig. 16 - Comparison between the curve of water-cut of the reference realization and that of realization 98.
Fig. 17 – Two consecutive realizations of the realization chain of the Boolean model.

Fig. 18 – Comparison between the curves of water-cut of realization 130, realization 131 and the reference realization.

Fig. 19 – Water saturation, after 10 years of injection and production.
Fig. 20 - Iterative calibration of the realization in Fig. 12(a): Realization calibrated after 11 iterations

Fig. 21 - Objective function versus the number of iterations

Fig. 22 - Comparison between the curve of water-cut of the reference realization and those of the initial realization, the realization after 2 iterations and the realization after 11 iterations.

Fig. 23 - Iterative calibration of the realization on Fig. 12(b): Realization calibrated after 4 iterations

Fig. 24 - Objective function versus the number of iterations

Fig. 25 - Comparison between the curve of water-cut of the reference realization and those of the initial realization, of the realization after 4 iterations.