Conditional simulations of reservoir models using Sequential Monte-Carlo methods

Alan Troncoso

PSL UNIVERSITÉ PARIS

THÈSE DE DOCTORAT

DE L'UNIVERSITÉ PSL

Préparée à MINES PARIS

Conditional simulation of reservoir models using Sequential Monte-Carlo methods

Simulations conditionnelles de modèles de réservoirs par méthodes séquentielles de Monte-Carlo

Soutenue par

Alan TRONCOSO

Le 21 Septembre 2022

École doctorale n°398

Géosciences, Ressources Naturelles et Environnement

Spécialité

Géosciences et géo-ingénierie

Composition du jury :

Chantal DE FOUQUET Directrice de recherche, MINES PARIS	*Présidente*
Guillaume CAUMON Professeur, Université de Lorraine	*Rapporteur*
Valérie MONBET Professeure, Université de Rennes 1	*Rapportrice*
Denis ALLARD Directeur de recherche, INRAE	*Examinateur*
Xavier EMERY Professeur, Université du Chili	*Examinateur*
Xavier FREULON Chargé de recherche, MINES PARIS	*Examinateur*
Jacques RIVOIRARD Retraité, MINES PARIS	*Directeur de thèse*

MINES PARIS | PSL

ABSTRACT

Extended Abstract

Over the last 20 years, a specific sequential Monte Carlo approach named particle filtering has been developed to perform prediction, filtering and smoothing for hidden-state Markov Chains. This method has been extended to a spatial context, which considerably increases its range of applicability in numerous fields of sciences, including the Earth Sciences. In this thesis, particle filtering is used to perform conditional simulations of reservoir models. It is applied to the Boolean model, an object-based model, and the Flumy model, a process-based model.

The Boolean model is a prototype of object-based models because objects do not interact. It can be used to model two-facies reservoirs, one facies composed of porous sand-bodies (foreground), and the other facies made of shales or clay that act as barriers to the circulation of fluids (background). It is mathematically tractable and its parameters can be statistically inferred. There exists an iterative algorithm to perform simulations of the Boolean model subject to point support conditions. However, the rate of convergence of this algorithm is difficult to establish. As an alternative, a sequential algorithm is proposed. This algorithm sequentially assimilates the conditioning foreground points using particle filtering. The conditioning background points are treated separately using a standard procedure. In complement, a method has been set up to quantify the quality of the conditional simulations thus produced. Overall, the sequential algorithm outperforms the iterative algorithm in terms of result quality and computing time.

Flumy is a process-based reservoir model designed for meandering channels in fluvial and turbiditic environments. The model simulates physical processes of channelized systems with stochastic parameters. The conditioning field data include well facies information, seismic maps, stratigraphic surfaces. In the original method used to produce conditional simulations, the processes are iteratively adapted by attracting or repelling the channels from their current location in order to match conditioning well data. The development of the conditioning method requires a deep understanding of the processes in order to modify them while avoiding undesirable artifacts or bias in results. For those reasons, another approach has been considered, which takes advantage of Flumy capacity to create unconditional simulations of reservoir models with high quality, in terms of reproduction of heterogeneity and connectivity. It consists of building the reservoir by stacking successive horizontal layers while assimilating the facies data using particle filtering. Additionally, this method can integrate other types of conditioning data and is easy to implement since the physical processes in the model do not need to be modified. However, the computational time and memory management are highly demanding. To compare both conditioning methods, a synthetic case and a real case, drawn from Loranca basin in Spain, are treated. A sensitivity analysis has permitted to identify the best parameters for the particle filtering algorithm. The results in terms of well matches and reservoir quality are promising. In the synthetic case, results are better than the original conditional simulation method in terms of

well facies match. However, regarding the real case, the original conditional simulation provides better results. This probably comes from the fact that the number of particles was reduced due to computational capacity restrictions.

In conclusion, a novel methodology based on particle filtering has been proposed to perform conditional simulations. This methodology is applicable to a variety of problematics. Two applications in simulations of reservoir models have been performed. The first one, the Boolean model, shows a perfect facies match and a short computational time. The second one, Flumy, shows promising results regarding the conditioning of process-based models.

Résumé étendu

Pendant les 20 dernières années, une approche séquentielle de Monte Carlo spécifique, appelée filtrage particulaire, a été développée pour effectuer la prédiction, le filtrage et le lissage dans le cas de chaînes de Markov cachées. Cette méthode a été étendue à un contexte spatial, ce qui accroit considérablement ses possibilités d'application dans de nombreux domaines des sciences, dont les Sciences de la Terre. Dans cette thèse, le filtrage particulaire a été utilisé pour simuler conditionnellement des modèles de réservoir. La méthode a été appliquée au schéma Booléen, un modèle basé sur des objets, et au modèle Flumy, un modèle basé sur les processus.

Le schéma Booléen est le plus simple des modèles basés sur des objets, car les objets n'interagissent pas entre eux. Le modèle peut être utilisé pour modéliser des réservoirs à deux faciès : un faciès composé par des corps sableux poreux (la "forme"), et un autre faciès fait d'argile qui agit comme barrière à la circulation des fluides (le "fond"). Le modèle se prête bien à des calculs mathématiques et ses paramètres peuvent être inférés statistiquement. Il existe déjà un algorithme itératif permettant de simuler un schéma Booléen sous des contraintes de support ponctuel. Cependant, le taux de convergence de cet algorithme est difficile à établir. À titre d'alternative, une méthode séquentielle est proposée. Cet algorithme séquentiel assimile les points conditionnants de la forme par l'usage du filtrage particulaire. Les points conditionnants du fond sont traitées séparément à travers une procédure standard. En complément, une méthode a été développée pour quantifier la qualité des simulations conditionnelles ainsi produites. Finalement, l'algorithme séquentiel surpasse l'algorithme itératif en termes de qualité de résultats et de temps de calcul.

Flumy est un modèle de réservoir basé sur les processus sédimentaires, créé pour des chenaux méandriformes en environnement fluvial ou turbiditique. Le modèle simule des processus physiques de systèmes chenalisés avec des paramètres stochastiques. Les données conditionnelles incluent des informations de faciès issus des puits, des cartes sismiques et des surfaces stratigraphiques. Dans la méthode originale utilisée pour la production de simulations conditionnelles, les processus sont adaptés itérativement, attirant ou repoussant le chenal à partir de sa position courante pour s'accorder aux données de puits conditionnantes. Le développement de cette méthode de conditionnement requiert une profonde compréhension des processus pour les modifier tout en évitant artefacts indésirables ou biais dans les résultats. Pour ces raisons, une autre approche a été considérée qui tire avantage de la capacité de Flumy à créer des simulations non-conditionnelles de haute qualité en termes de reproduction d'hétérogénéité et de connectivité. La méthode consiste à construire le réservoir en empilant des couches horizontales successives pendant qu'elle assimile leurs données de facies en utilisant le filtrage particulaire. De plus, cette méthode peut intégrer d'autres types de données conditionnelles. Elle est également facile à implémenter, car les processus physiques du modèle n'ont pas besoin d'être modifiés. Cependant, elle est très gourmande en temps calcul et en mémoire. Pour comparer ces deux méthodes de conditionnement, un cas synthétique et un cas réel (bassin de Loranca, en Espagne) ont été traités. Une analyse de sensibilité a permis d'identifier les meilleures valeurs de paramètres pour l'algorithme de filtrage particulaire. Les résultats en termes de reproduction des faciès aux puits et de qualité de réservoir sont prometteurs. Dans le cas synthétique, la méthode permet une meilleure reproduction des faciès aux puits que la méthode originale de simulation conditionnelle.

En revanche, dans le cas réel, elle donne de moins bons résultats. Cet effet vient, probablement, du fait que le nombre de particules a dû rester faible à cause de restrictions informatiques.

Pour conclure, une nouvelle méthodologie basée sur le filtrage particulaire a été proposée pour fournir des simulations conditionnelles. La méthodologie est applicable à une variété de problématiques. Deux applications dans le cas de simulation de modèles de réservoir ont été faites. Dans le premier, le schéma Booléen, elle montre une correspondance parfaite des faciès et un temps de calcul court. La seconde, sur l'exemple de Flumy, montre des résultats prometteurs pour conditionner les modèles basés sur les processus.

DEDICATION

En esta página, quisiera agradecer a todos quienes fueron en este trabajo, partícipes durante los casi cuatro años que Francia me abrió sus puertas para realizar mi doctorado.

A mi familia, que fueron incondicionales desde el inicio con la decisión de partir al extranjero a esta aventura. Padres, tíos, abuelos y primo: gracias por su crianza, valores y moral que me ayudaron sobremanera a sortear esta prueba y por ser parte del proceso de aprendizaje desde el comienzo de mi vida. Su apoyo primó durante todo el proceso, siendo aún mayor en la época de pandemia, cuando la preocupación acrecentaba.

A mis amigos de Chile que el lazo no ha sido cortado por el tiempo. Estargio, Robomba, Pachao, Salvapuerta, Mario, María Camila, Jamirer, Xonpa, CZ, Morales, Jay, Cata, Diago, Bonno, Lefonse, Camila, Karollo, Seviene y otros. Fueron el vínculo para conservar el castellano chileno y seguir las historias de nuestro largo y angosto país.

Je voudrais remercier à l'équipe de l'école de mines qui m'ont accueilli de la meilleure manière dès que je suis arrivé.

L'équipe de geostats et en général : Jacques RIVOIRARD (directeur de thèse), Xavier FREULON (professeur encadrant), Christian LANTUÉJOUL et Fabien ORS (membres de mon équipe de travail), Didier RENARD, Chantal DE FOUQUET, Hans WACKERNAGEL, Thomas ROMARY, Emilie CHAUTRU, Nicolas DESASSIS, Mike PEREIRA, Natalie DIETRICH, Gaëlle LE LOC'H, Véronique LACHASSE, Sandrine MOTTE, etc.

Les membres du comité de suivi de thèse, et les rapporteurs et les examinateurs de ma thèse : Valérie MONBET, Guillaume CAUMON, Denis ALLARD, Xavier EMERY et Simon LOPEZ.

Les amis français (et étrangers): Zhuldyz, Juanito, Marine, Léa, Nicolas Señor, Antoine, Irina, Alexis, Charly, Ferdinand, Aureliens, Kirilova, Laure, Carole, Nicolò, Lucia, Wafa, Helena, Shifu Jinling, Neo, etc.

À la résidence universitaire, qui m'a donné le logement et l'opportunité d'être l'ambassadeur durant la pandémie de la COVID-19: Catherine, Claude, Amélie et les différents résidents.

A los amigos latinos en Fontainebleau que estuvieron y están siempre presentes en las aventuras por el mundo: la Sabrina, la Katherine, la Sarai, el Eric, la Gaby, la Andrea, la Yasmeen, el Diego, y otros.

Finalmente, a ANID por permitirme estudiar en Francia a través de la beca BECAS DE DOCTORADO EN EL EXTRANJERO BECAS CHILE – CONVOCATORIA 2018.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

La modélisation de réservoirs est une tâche importante en géosciences. Cette activité demande de comprendre le contexte de formation du réservoir et de reproduire son architecture. Souvent, les données disponibles sont peu abondantes et sont dispersées dans l'espace, raisons pour lesquelles on choisit des modèles spatiaux stochastiques. La construction de simulations conditionnelles aux données permet en outre de mesurer les fluctuations statistiques de ces modèles. Dans ce travail de thèse, deux modèles sont utilisés pour représenter les réservoirs : le schéma Booléen et le modèle Flumy pour des systèmes chenalisés méandriformes. Ce travail de thèse développe des algorithmes de simulations conditionnelles basés sur une technique statistique appelée filtrage particulaire.

At the outset, there is the notion of reservoir. A *reservoir* is a subsurface domain made of porous and permeable rock bodies that hold fluids of interest (such as hydrocarbons, dissolved uranium during leaching, hot water or CO_2), enclosed within impermeable rocks that act as flow barriers. To recover or store these fluids, it is critical to acquire information about the flow behaviour within the reservoir, which is usually performed by flow simulations. For example, in the hydrocarbon industry, water and gas are injected in the reservoir to increase the fluid pressure and allow hydrocarbons extraction using production wells (Lake, 1989). Similarly, in the uranium industry, acids are injected to leach the uranium ore that is recovered in solution using production wells (Collet et al., 2022). In the geothermal energy industry, the hot water is used as a carrier to transfer the heat up to accessible depths and transform it into electricity (Barbier, 2002). One of the most effective ways to reduce greenhouse gas emission is to store CO_2 in geologic bodies, for example oil and gas reservoirs. Oil reservoirs are much better explored and developed than other geologic bodies, with clearer characteristics and more data available. The potential of CO_2 storage in a reservoir must be evaluated to determine its maximum theoretical and effective capacity (Pingping et al., 2009).

Among the different factors that affect the circulation of fluids in a reservoir, its architecture plays a prominent role. This architecture describes, the shape and the orientation of the different bodies with their facies, as well as the way they are put together. Understanding the architecture is the first task of the geological modelling of a reservoir. Another important task is to know the context of the reservoir formation, that is, the system in which the reservoir was formed. Unfortunately, this is not always comprehensive as knowledge on both the architecture and the context, depend mainly on the field data. This data can be classified as hard or soft as a function of its degree of certainty. Examples of hard data are well data. Such data are usually scarce and expensive. Examples of soft data include not only interpreted seismic data and other geophysical data, but also outcrops or analogue reservoirs. Soft data are more abundant and cheaper.

Most often, the available data is scarce, which entails that there can exist several physical realities fully compatible with the data. In order to cope with this problem, the approach adopted in

this thesis is to model the reservoir architecture using a *spatial stochastic model*. Within this framework, the data is used to perform the statistical inference of the model parameters. It is also employed to predict the statistical fluctuations of the model that remain once data has been taken into account.

Regarding this second objective, note that it cannot be met using standard *kriging techniques* (Armstrong, 1984; Chilès and Delfiner, 1999; Matheron, 2019)[1] because they are not strictly designed to accommodate soft data. Moreover, kriging results often exhibit undesired smoothing properties. Finally, the evaluation of the uncertainty in terms of kriging variance is not appropriate.

Another technique is the construction of *conditional simulations*, that are realizations of the spatial stochastic model compatible with the available data. The uncertainty associated with this conditional model is provided by the statistical fluctuations observed between conditional simulations.

Suppose that a reservoir architecture model has been chosen and its parameters statistically inferred. The objective of this thesis is to design algorithms for conditionally simulating the model. Of course, the algorithm to be designed will depend on the available data (people sometimes speak of data integration) and also of the model considered.

Among the spatial stochastic models, one can distinguish between

- Spatial random field models that have their statistical properties characterized by a spatial distribution. Typical examples are the Gaussian excursion random sets, or more generally, the plurigaussian random fields (Armstrong *et al.*, 2011; Emery, 2007);
- Object-based models that can be seen as populations of interacting random objects. These interactions may entail not only the spatial distribution of the object locations, but also the size and shape dependence relationships between objects. A prototype of such models is the Boolean model that is made of independent random objects (chapter 3);
- Process-based models that imitate sedimentation processes in order to progressively build a reservoir. As an example, Flumy is designed to model reservoirs formed by meandering channelized systems (chapter 4).

To conditionally simulate spatial random field models or object-based models, there exist iterative Markovian algorithms, such as the Metropolis-Hastings algorithm or the Gibbs sampler, that simultaneously handle all data at each iteration (chapter 2). Unfortunately, they converge very slowly. In this thesis, a Monte Carlo type algorithm called *particle filtering* is implemented and tested. It allows a *sequential assimilation of conditioning data*.

To conditionally simulate process-based models, there already exists an ad-hoc conditional algorithm for Flumy. We will compare it with a sequential algorithm based on particle filtering that can be applied provided that an aggradation assumption holds true. This assumption allows us to keep the reservoir consolidated while it is built.

To sum up, the objective of this thesis is to examine how particle filtering can be used to produce conditional simulations of some spatial stochastic models of reservoirs. The models considered here are assumed to be known, so inference issues are not considered.

This thesis is organized as follows. Chapter 2 provides some background on the way to simulate distributions using Monte-Carlo methods. This constitutes the basis for the conditional simulation of stochastic reservoir models. A comparison is made between iterative and sequential methods, and examples of algorithms are given. Chapter 3 deals with the Boolean model. It starts with the specification of this model, a brief review of its properties and the description of an algorithm to simulate it unconditionally. Then, two conditional simulation algorithms are

[1] The book by Matheron is the English translation of a course given in the early seventies at the School of Mines of Paris

presented. Whereas the iterative one, based on the Metropolis-Hastings algorithm is standard, the sequential one, hinged on particle filtering, is novel. To compare their performances, new criteria have been developed. Chapter 4 is devoted to Flumy. After a general presentation of this process-based model, the current conditional simulation algorithm is described. An alternative is proposed that rests on the particle filtering. Both algorithms are first illustrated in a synthetic case study. Their performances are also demonstrated on a real case study. To compare them, well facies matches and global sand proportions are resorted to. Finally, chapter 5 draws conclusions and gives some perspectives to this work.

2

SIMULATION OF DISTRIBUTIONS USING MONTE-CARLO METHODS

Le filtrage particulaire, une technique statistique séquentielle de Monte Carlo, a été développé ces dernières années pour effectuer la prédiction, le filtrage et le lissage de distributions dans le cas de chaînes de Markov cachées. Dans ce travail de thèse, une autre application du filtrage particulaire a été conçue pour la simulation conditionnelle de modèles stochastiques spatiaux, dans le cadre spécifique de la modélisation de réservoirs. L'aspect théorique de la simulation itérative plus classique et celui de la simulation séquentielle sont décrits et illustrés par des exemples.

2.1 Introduction

In chapters 3 and 4, we will be faced with the problem of simulating multivariate distributions, the components of which are subject to conditions or constraints. These distributions have usually a probability density function (p.d.f.) that takes the general form

$$f(x) = \alpha_k \, \phi(x) \prod_{i=1}^{k} h_i(x_i) \qquad x = (x_1, ..., x_k) \in \mathbb{R}^k. \tag{2.1}$$

In this formula, $\phi(x)$ is another p.d.f., and $h_1, ..., h_k$ are non-negative functions that specify the constraints upon each component of f. A typical example is $h_i(x_i) = \mathbb{1}_{x_i \in A_i}$ that indicates that the component x_i belongs to the domain A_i. The coefficient α_k is a normalization constant.

In what follows, the set $\{x \in \mathbb{R}^k : f(x) > 0\}$ that supports f will be denoted by S.

The complex expression of those distributions precludes the simple use of standard algorithms such as inversion or acceptance-rejection algorithms (Devroye, 1986; Rubinstein, 1981), and suggests instead to resorting to Monte Carlo algorithms. This chapter gives some background to two of them, referred to as *Markov chain Monte Carlo* and *Sequential Monte Carlo algorithm*. The first one is iterative and the second sequential. Both algorithms will be presented in turn.

2.2 Iterative algorithms

2.2.1 Background

The algorithm consists of simulating a sequence of random states $(X_n)_{n \in \mathbb{N}}$ in such a way that the distribution of X_n tends to f as n becomes very large. Its implementation is largely facilitated if the following assumptions are made:

1. Markov property For each $n > 1$, the distribution of X_n given $X_1, ..., X_{n-1}$ depends solely on X_{n-1}. It is specified by a transition kernel $P_n(x, A) = P\{X_n \in A \mid X_{n-1} = x\}$

for each state $x \in S$ and each Borelian $A \in \mathcal{B}(S)$. This avoids keeping into memory at each iteration the whole sequence of states previously generated.

2. Homogeneity The device that allows to jump from a state to the next one does not depend on the iteration order. In other words, the transition kernel P_n is the same at all iterations.

Starting from such assumptions, the ingredients required for simulating the sequence of random states are the distribution f_0 of the initial state and a transition kernel P that tells how to jump from a current state to the next one. More generally, the probability of transiting from x to A in n steps is $P^n(x, A)$ where P^n is the n^{th} iterate of P.

Now, the question comes as how to design these two ingredients to simulate the target distribution f? An answer is given by a classical theorem of the Markov chain theory (Meyn and Tweedie, 1993; Tierney, 1994): for any state x and any Borelian A, $\lim_{n\longrightarrow\infty} P^n(x, A) = f(A)$ as soon[1] as the transition kernel P satisfies the following three properties:

1. Irreducibility. For any $x \in S$ and any $A \in \mathcal{B}(S)$, there exists $n > 0$ such that $P^n(x, A) > 0$.

2. Aperiodicity. For any $x \in S$ and any $A \in \mathcal{B}(S)$, the greatest common denominator of the n's such that $P^n(x, A) > 0$ is 1.

3. Invariance. For any $A \in \mathcal{B}(S)$, we have

$$\int_S f(x)\, P(x, A)\, dx = f(A)$$

2.2.2 Three examples

2.2.2.1 Metropolis-Hastings algorithm

The objective of this algorithm imagined by Metropolis et al. (1953) and generalized by Hastings (1970) is to design a transition kernel for simulating f. It requires an auxiliary transition kernel Q on $S \times \mathcal{B}(S)$, supposed here to admit a density

$$Q(x, A) = \int_A q(x, y)\, dy.$$

Starting from a current state x, the idea is to use Q to generate a candidate state y. This state is subsequently accepted with probability $\alpha(x, y)$ or rejected with the complementary probability $1 - \alpha(x, y)$, in which case the current state x is maintained. Here is the generic algorithm for one iteration[2]

Algorithm 1: One iteration of Metropolis-Hastings algorithm

Input: x current state
Output: y next state
generate $y \sim q(x, \cdot)$ and $u \sim \mathcal{U}$;
if $u \geq \alpha(x, y)$, **then put** $y = x$

The transition kernel P produced by this algorithm can be written

$$P(x, A) = \int_A q(x, y)\, \alpha(x, y)\, dy \;+\; r(x)\, \mathbb{1}_{x\in A} \qquad x \in S,\ A \in \mathcal{B}(S) \tag{2.2}$$

[1] $f(A)$ is a short notation for $\int_A f(x)\, dx$.
[2] For the generation of samples from the uniform distribution, see L'Ecuyer (2017)

with

$$r(x) = \int_S q(x,y)\,[1-\alpha(x,y)]\,dy \qquad x \in S.$$

There must be conditions on Q and α for P to satisfy the three properties mentioned above. Let us consider them in turn.

1. Irreducibility. It is well known that the irreducibility of Q (for f) is necessary, but generally not sufficient to warrant that of P (Tierney, 1994). Further conditions are therefore required. Robert and Casella (2004) propose to take $q(x,y) > 0$ for all $x, y \in S$. Roberts and Smith (1994) recommend the less drastic condition $q(x,y) = 0$ if and only if $q(y,x) = 0$ for all $x, y \in S$;

2. Aperiodicity. Suppose P is irreducible. Then P will be aperiodic provided that the Markov chain has a chance to sojourn in the same state during successive iterations. Tierney (1994) formulates this event by putting $f(R) > 0$, where $R = \{x \in S : r(x) > 0\}$;

3. Invariance. This property is trivially satisfied as soon as P fulfils the reversibility property

$$\int_A f(x)\,P(x,B)\,dx = \int_B f(x)\,P(x,A)\,dx \qquad A, B \in \mathcal{B}(S) \tag{2.3}$$

(take $B = S$). Owing to the definition of P, this reversibility property is equivalent to

$$\int_A \int_B f(x)\,q(x,y)\,\alpha(x,y)\,dxdy = \int_B \int_A f(y)\,q(y,x)\,\alpha(y,x)\,dydx. \tag{2.4}$$

It suffices to design α so that equality (2.4) is satisfied. The standard choice proposed by Hastings (1970) is

$$\alpha(x,y) = \min\left(\frac{f(y)\,q(y,x)}{f(x)\,q(x,y)}, 1\right),$$

but other choices are possible, such as that of Barker (1965):

$$\alpha(x,y) = \frac{f(y)\,q(y,x)}{f(x)\,q(x,y) + f(y)\,q(y,x)}.$$

As an illustration, take for f the distribution given in (2.1). If $\phi > 0$ on $\mathbb{R}^k$, then the simulation of f can be achieved by taking the auxiliary transition kernel $q(x,y) = \phi(y)$ (independent sampler strategy). Hastings' standard choice leads to the acceptance probability

$$\alpha(x,y) = \min\left(\prod_{i=1}^{k} \frac{h_i(y_i)}{h_i(x_i)}, 1\right).$$

2.2.2.2 Restriction of a Markov chain

The aim of this section is to show a device proposed by Lantuéjoul (2002) to turn an unconditional algorithm into a conditional one. Let f be a distribution defined on a particular state space (there is no inconvenience to denote it by S). Suppose that the insertion of new conditions or constraints excludes certain states so that only the states within a subset S_c of S are allowed. For simplicity, we assume $f(S_c) > 0$. Let f_c be the conditional distribution induced by f on S_c:

$$f_c(x) = \frac{f(x)}{f(S_c)} \qquad x \in S_c.$$

The objective is to simulate f_c. More precisely, assume that f can be iteratively simulated using a transition kernel P. Can this help to simulate f_c?

A natural approach consists of running P from S_c and forbidding any excursion outside S_c. This leads to consider the restricted transition kernel

$$P_c(x, A) = P(x, A) + \mathbb{1}_{x \in A}\, P(x, S \backslash S_c) \qquad x \in S_c,\ A \in \mathcal{B}(S_c) \tag{2.5}$$

It can be noticed that this formula can be retrieved from (2.2) by taking $Q(x, dy) = P(x, dy)$ and $\alpha(x, y) = \mathbb{1}_{y \in S_c}$. Accordingly, the algorithm produced by this approach is nothing but a particular case of a Metropolis-Hastings algorithm.

The properties of this restricted transition kernel are as follows. P_c is not always irreducible as this strongly depends on S_c. This must be tested on a case-by-case basis. P_c is aperiodic if

$$\int_{S_c} f(x)\, P(x, S \backslash S_c)\, dx > 0 \tag{2.6}$$

Finally P_c leaves f_c invariant if it is f_c-reversible, which is true if P is itself f-reversible.

It follows that if P_c is irreducible, and if P is f-reversible and satisfies (2.6), then the transition kernel P_c can be used to simulate f_c.

2.2.2.3 Gibbs sampler

Popularized by Geman and Geman (1984), the Gibbs sampler can be seen as a stochastic version of the Gauss-Seidel algorithm that was developed to solve linear systems of equations. Here is a simple version of it. At each iteration, one component of the current state is picked up at random and updated according to the target distribution conditionally on the other components. In the following algorithm, x_{-i} denotes the vector obtained by suppressing the i^{th} component of x. $f(\cdot \mid x_{-i})$ stands from the distribution of the i^{th} component given x_{-i}.

Algorithm 2: One iteration of the Gibbs sampler

Input: $x = (x_1, ..., x_k)$ current state
Output: $y = (y_1, ..., y_k)$ next state
generate $i \sim \mathcal{U}(\{1, ..., k\})$;
put $y_{-i} = x_{-i}$ and **generate** $y_i \sim f(\,\cdot \mid x_{-i})$;

One can readily see that the transition kernel associated with the Gibbs sampler is

$$P(x, A) = \frac{1}{k} \sum_{i=1}^{k} \int_{\mathbb{R}} f(u \mid x_{-i})\, \mathbb{1}_{(x_{-i}, u) \in A}\, du$$

It remains to specify the range of validity of this algorithm. This is done by making explicit conditions to ensure that the irreducibility, aperiodicity and invariance properties are satisfied by P.

Irreducibility: P is irreducible if and only if S cannot be written $S' \cup S''$ with S' and S'' separated by hyperplanes orthogonal to the coordinate axes. The following diagram shows two examples of supports.

If S is connected, then P is always irreducible. In the particular case of the distribution (2.1), if ϕ is strictly positive on $\mathbb{R}^k$, then P is also irreducible, whatever the h functions.

Aperiodicity: P is aperiodic because the same random component can be selected at two successive iterations.

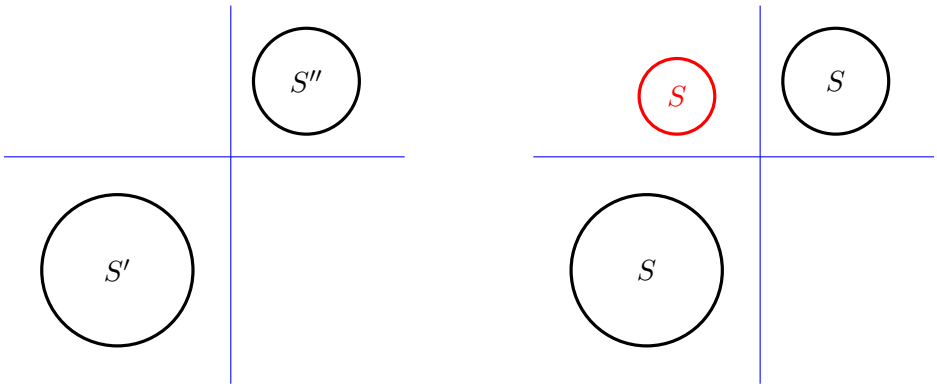


Figure 2.1: Irreducibility schema. On the left, the schema leads to a reducible transition kernel. On the right, one to an irreducible one.

Invariance: the proof is straightforward:

$$\begin{aligned}\int_S f(x)\,P(x,A)\,dx &= \int_S f(x)\,\frac{1}{k}\sum_{i=1}^{k}\int_{\mathbb{R}} f(\,t\mid x_{-i})\,\mathbb{1}_{(x_{-i},u)\in A}\,du\,dx\\ &= \frac{1}{k}\sum_{i=1}^{k}\int_S f(x_{-i},t)\int_{\mathbb{R}} f(u\mid x_{-i})\,\mathbb{1}_{(x_{-i},u)\in A}\,du\,dx_{-i}\,dt\\ &= \frac{1}{k}\sum_{i=1}^{k}\int_S f(t\mid x_{-i})\int_{\mathbb{R}} f(x_{-i},u)\,\mathbb{1}_{(x_{-i},u)\in A}\,du\,dx_{-i}\,dt\\ &= \frac{1}{k}\sum_{i=1}^{k}\int_S f(t\mid x_{-i})\,f(A)\,dt = f(A).\end{aligned}$$

The Gibbs sampler can be used to simulate any multivariate distribution provided that its full conditional distributions can be easily simulated. In the case of the distribution (2.1), this boils down to simulating the univariate densities $x_i \mapsto f(\,\cdot\mid x_{-i})\,h_i(x_i)$ for each $i = 1, ..., k$.

The three preceding examples suggest that the design of a transition kernel is generally not a difficult issue. More critical is the *rate of convergence* of the Markov chain, that quantifies the evolution of the difference between the current and the stationary distributions versus the number of iterations. In practice, the first generated states are often discarded to smooth away the influence of the initial states (the so called *burn-in period*). This does not prevent the fact that the Markov chain may sojourn in metastable states of equilibrium for long periods. In the case of the Metropolis algorithm, Morgensen and Tweedie (1996) have established that the rate of convergence is geometric. Regarding the Markov chain built starting from the restriction principle, its rate of convergence must be studied on a case-by-case basis. As a rule of thumb, the iterations are often stopped by the practitioners when certain conditions are met. Several examples will be shown in the next chapters.

2.3 Sequential algorithms

Whereas iterative algorithms update all components of f simultaneously, sequential algorithms simulate one component of f after the other, in a prespecified order. This section is devoted to the powerful *sequential Monte Carlo methods* (Doucet et al., 2001; Del Moral, 2004; Kroese et al., 2013). They rest on two basic algorithms, importance sampling and resampling, that are presented first.

Algorithm 3: Simulation using importance sampling

Input: Target distribution: f
Input: Sampling distribution: g
Input: Number of drawings: p
Output: Simulated vector: x
for $j = 1 : p$ **do**
  generate $x^{(j)} \sim g$;
  set $w^{(j)} = f(x^{(j)})/g(x^{(j)})$;
end
generate $j \sim \mathcal{D}(w)$;
return $x = x^{(j)}$

2.3.1 Importance sampling

Suppose that we want to simulate a p.d.f. f defined on $\mathbb{R}^d$. This algorithm relies on another p.d.f. g that is also defined on $\mathbb{R}^d$ and that dominates f, in the sense that $g(x) = 0$ implies $f(x) = 0$. The idea is to produce independent samples $x^{(1)}, ..., x^{(p)}$ from g, and to assign each one a weight that is the likelihood ratio between f and g

$$w^{(j)} = \frac{f(x^{(j)})}{g(x^{(j)})} \qquad j = 1, ..., p$$

It then remains to draw an index j from the normalized weight distribution. The returned value is $x^{(j)}$. Algorithm 3 makes the approach more explicit[3].

Regarding this algorithm, a pending question is how many drawings must be performed to get an approximate simulation with a prespecified precision. It turns out that the result depends on the expected likelihood ratio

$$\rho = \int \frac{f(x)}{g(x)} f(x)\, dx$$

that measures the adequacy between the target distribution and the importance distribution. Let $\dot{f}_p$ be the probabilistic version of $\sum_{j=1}^{p} w^{(j)} \delta_{x^{(j)}}$. It has been established by Agapiou et al. (2017) and Sanz-Alonso et al. (2018) that if $\rho > 0$, then

$$|\mathbb{E}_g[\dot{f}_p(\varphi) - f(\varphi)]| \leq \frac{2\rho}{p}$$
$$\mathbb{E}_g[\dot{f}_p(\varphi) - f(\varphi)]^2 \leq \frac{4\rho}{p}$$

for each test function φ such as $|\varphi| \leq 1$. It turns out that the convergence is inversely proportional to p.

This simulation algorithm is especially efficient in low-dimensional workspaces. When d increases, it becomes more and more difficult to find an importance density that can be easily simulated and provides a small expected likelihood ratio.

2.3.2 Resampling

Let $\mathcal{D}(w)$ be a discrete distribution on $\{1, ..., p\}$ with $w^{(1)} + ... + w^{(p)} = 1$. Let also m be a positive integer. The problem considered here is the design of p random integer variables $N^{(1)}, ..., N^{(p)}$

[3] In this algorithm, $\mathcal{D}(w)$ denotes the discrete distribution that takes the value $1, ..., p$ with respective probabilities

$$p^{(j)} = \frac{w^{(j)}}{\sum_{j'=1}^{p} w^{(j')}} \qquad j = 1, ..., p.$$

such that $\sum_{j=1}^{p} N^{(j)} = m$ and

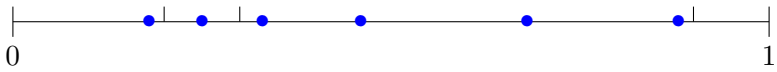
$$\mathbb{E}\{N^{(j)}\} = m\, w^{(j)} \qquad j = 1, ..., p$$

To achieve this goal, various algorithms have been proposed. A comparison of the main algorithms often used can be found in Douc et al. (2005) and Chopin and Papaspiliopoulos (2020), a summary of which is given here. It is illustrated with $p = 4$ and $m = 6$.

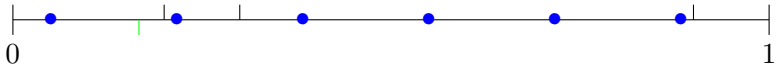
Suppose that the weights of w are ordered according to the following scheme:

In the following four examples, $N^{(J)}$ is the number of random points fallen in the interval specified by $w^{(j)}$.

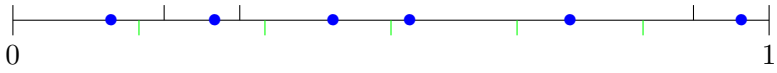
- Multinomial resampling The m points are independent and uniform on $[0, 1]$.



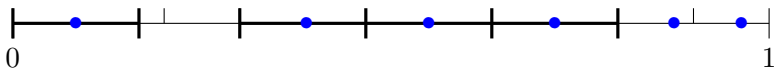
- Systematic resampling The m points are regularly arranged starting from an offset uniform on $[0, 1/m]$.



- Stratified resampling The segment $[0, 1]$ is split into m intervals of length $1/m$. One point is uniformly generated within each interval.



- Residual resampling At first each weight interval is assigned a number of points equal to the number of sub-interval of length $1/m$ it contains. Then the remaining points are independently and uniformly located in the residual space.



Among those four algorithms, the multinomial and the residual resamplings are preferred because their results are not sensitive to the weight order (Douc et al., 2005). Note however that if the weight order is privileged, then the weights can always be reordered by random permutation of their labels. More important is the fact that, in an estimation perspective, the residual and the stratified resamplings tend to provide results with the smallest variance. In the case of the residual resampling, this may be related to the fact that if $m \geq p$, then the longest weight interval is never devoid of points. Chopin and Papaspiliopoulos (2020) discourage the use of the multinomial resampling as its performances are outperformed by the other resampling algorithms.

We close this section with a few words on notations. Consider the p items $x^{(1)}, ..., x^{(p)}$ and suppose they are affected with the (not necessarily normalized) weights $w^{(1)}, ..., w^{(p)}$. If their resampling yield the new items $y^{(1)}, ..., y^{(m)}$, then we write

$$y^{(1:m)} = \mathbf{resampling}(x^{(1:p)}, w^{(1:p)}),$$

irrespective of the resampling option. In this expression, the superscript '$1:p$' has been introduced. It stands for the *set* $\{1,...,p\}$. It should not been confounded with '$\leq p$' that is the *ordered set* $(1,...,p)$. Another useful notation for subscripts or superscripts is '$<p$'. It stands for the ordered set $(1,...,p-1)$ if $p>1$ and $\varnothing$ otherwise.

2.3.3 Sequential importance sampling with resampling

The problem addressed here is the simulation of the distribution given in (2.1). For what follows, it is convenient to write it like

$$f(x_{\leq k}) = \alpha_k\, \phi(x_{\leq k}) \prod_{j\leq k} h_j(x_j).$$

The decomposition of f into a product of univariate conditional p.d.f.'s is:

$$f(x_{\leq k}) = \prod_{1\leq i\leq k} f(x_i \mid x_{<i}).$$

For $k=1$, $f(x_1 \mid x_{<1}) = f(x_1)$ suggests simulating the components of f sequentially. At the step i, the component of order i is simulated. However, this idea is difficult to implement because the expression of these univariate p.d.f.'s is quite intricate[4]:

$$\begin{aligned}
f(x_1 \mid x_{<1}) &= h_1(x_1) \int \prod_{i'>1} h_{i'}(u_{i'})\, \phi(x_1, u_{>1})\, du_{>1} \\
f(x_i \mid x_{<i}) &= \frac{h_i(x_i) \displaystyle\int \prod_{i'>i} h_{i'}(u_{i'})\, \phi(x_i, u_{>i} \mid x_{<i})\, du_{>i}}{\displaystyle\int \prod_{i'\geq i} h_{i'}(u_{i'})\, \phi(u_{\geq i} \mid x_{<i})\, du_{\geq i}} \qquad 1<i<k \\
f(x_k \mid x_{<k}) &= \frac{h_k(x_k)\, \phi(x_k \mid x_{<k})}{\displaystyle\int h_k(u_k)\, \phi(u_k \mid x_{<k})\, du_k}
\end{aligned}$$

This comes from the fact that the simulation of each component requires not only the values taken by the components lower order, but also the constraints prescribed to the components of upper (this is precisely the incorporation of these constraints that gives rise to the presence of integrals in the formulas). To summarize, a direct sequential simulation can be hardly implemented.

To reduce the difficulties raised by these integrals, it is tempting to *forget* the constraints of the upper components. This leads to simulating the first component using the p.d.f.[5]

$$f_1(x_1) = \frac{\phi(x_1)\, h_1(x_1)}{\displaystyle\int \phi(u_1)\, h_1(u_1)\, du_1},$$

and the other components by induction using the conditional p.d.f.[6]

$$f_i(x_{\leq i} \mid x_{<i}) = \frac{\phi(x_i \mid x_{<i})\, h_i(x_i)}{\displaystyle\int \phi(u_i \mid x_{<i})\, h_i(u_i)\, du_i}.$$

Unfortunately, this approach does not yield satisfactory results. The reason lies in that it cannot anticipate the constraints to come. As an example, suppose that the constraints $x_i \in A_i$ and $x_{i+1} \in A_{i+1}$ hold. Suppose also that there exist $B_i \subsetneq A_i$ such that $x_i \in B_i$ implies $x_{i+1} \notin A_{i+1}$. In that case, the simulation of x_{i+1} is impossible and the algorithm stops without returning a result.

[4] These distributions correspond to smoothing distributions in the terminology of particle filtering (Chopin and Papaspiliopoulos, 2020).

[5] This distribution and the following correspond to filtering distributions in the terminology of particle filtering (Chopin and Papaspiliopoulos, 2020).

[6] In these formulas, $\phi(x_i)$ and $\phi(x_i \mid x_{<i})$ are the marginals and the conditional marginals of ϕ.

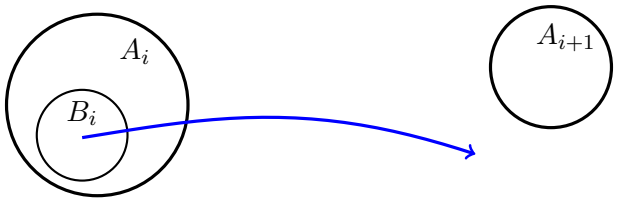


Figure 2.2: Scheme showing difficulties of not considering the future component of a sequential simulation. In the current state i, if $x \in B_i$, $x \notin A_{i+1}$, which is not allowed to continue the method.

A simple way to alleviate this problem, if not to sort it out, is to simulate each component by importance sampling. By replicating the x_i's values, the domain A_i is better explored. Several sequences $x_{\leq i}$ have no corresponding component x_{i+1} in A_{i+1}, in which case their weight vanishes. The others have a positive weight.
More generally, the constraints are not always of hit-or-miss type. The weight assigned to each sequence quantifies the adequacy between the sequence and the set of constraints already encountered.

Once a step has been performed, it often appears that most weights peter out whereas a few of them blow out. In order to compensate for such a degeneracy, a resampling procedure is prescribed at each step of the algorithm. This tends to privilege the particles that are in good adequacy with the constraints and discard the unlikely ones. At the end of the resampling procedure, the different sequences can be considered as equiprobable.

This approach is summarized in algorithm 4 that is presented using the terminology of particle filtering (Del Moral, 2004). The partial sequences of components are called particles and denoted by $x_{\leq i}^{(j)}$ where i ranges from 1 to k and j from 1 to p.

Algorithm 4: Sequential Simulation using importance sampling

Input: Target distribution: f
Input: Importance distribution: g
Input: Number of particles: p
Output: Simulation of f: $x_{\leq k}$
for $i = 1 : k$ **do**
  for $j = 1 : p$ **do**
    generate $x_i^{(j)} \sim g(\cdot \mid x_{<i}^{(j)})$;
    set $w_i^{(j)} = \dfrac{\phi(x_i^{(j)} \mid x_{<i}^{(j)})}{g(x_i^{(j)} \mid x_{<i}^{(j)})} h_i(x_i^{(j)})$;
  end
  if $\sum_{j=1}^{p} w_i^{(j)} = 0$ **then stop**;
  $x_i^{(1:p)} =$ **resampling** $\left(x_i^{(1:p)}, w_i^{(1:p)}\right)$;
end
generate $j \sim \mathcal{U}\{1, ..., p\}$;
return $x_{\leq k} = x_{\leq k}^{(j)}$.

As an illustration, let us show how to perform conditional simulations of an excursion set of a Gaussian random field. Consider the image on the left of Figure 2.3. This is a realization of a Gaussian random field with Gaussian covariance function (scale factor 10) in a domain 300×200. This simulation has been thresholded at level 0.5, and 100 points have been randomly selected to serve as conditioning data points (see right of Figure 2.3). Assuming that the model parameters

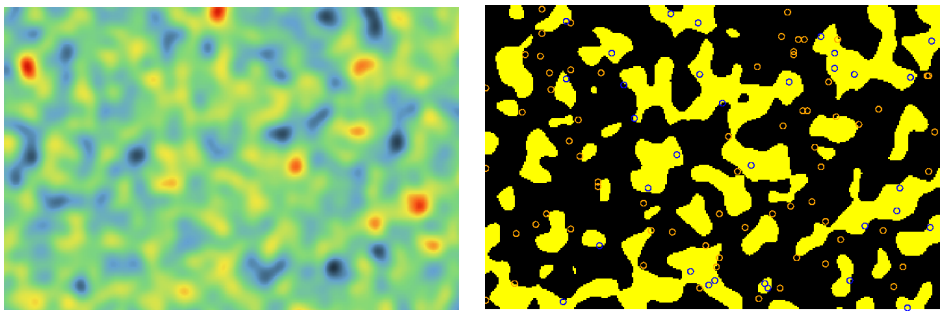


Figure 2.3: Simulation of a Gaussian random field (left) and one of its excursion sets used to build a population of conditioning data points (right)

(covariance function and threshold level) are known, how can we generate realizations of the model that reproduces the values (0 or 1) at all conditioning data points?

Such a conditional simulation exercise can be performed in three phases, namely (i) random assignment of a Gaussian value at each conditioning data point, (ii) conditional simulation of a Gaussian random field, and (iii) threshold of the Gaussian simulation at the specified level. Since the second phase is standard (Chilès and Delfiner, 1999) and the third phase is trivial, this chapter focuses on the first phase.

This first phase consists of simulating the distribution (2.1), where ϕ is a multivariate Gaussian p.d.f. and $h_i(x_i) = \mathbb{1}_{x_i \in A_i}$ with $A_i =]-\infty, 0.5[$ or $[0.5, +\infty[$ for each $i = 1, ..., k$. It is usually carried out using the Gibbs sampler (Chilès and Delfiner, 1999; Armstrong *et al.*, 2011; Lantuéjoul, 2002; Troncoso et al., 2017). Freulon (2013) recommended to use particle filtering as a substitute. Several choices are possible regarding the definition of importance density:

- if $g(x_i \mid x_{<i}) = \phi(x_i \mid x_{<i})$, then $w_i = \mathbb{1}_{x_i \in A_i}$;
- if $g(x_i \mid x_{<i}) \propto \phi(x_i \mid x_{<i})\, \mathbb{1}_{x_i \in A_i}$, then $w_i = \phi(A_i \mid x_{<i})$, which can be easily assessed by simple kriging.

The second choice has been preferred as the weights never vanish. Figure 2.4 shows 4 conditional simulations obtained using 500 particles. Despite the non-negligible number of data points, these simulations exhibit a large variety in terms of their geometry and their topology. Since the range of the Gaussian random field is fairly small, the influence of each conditioning data point is limited.

2.4 Conclusions

Multivariate distributions subject to conditions or constraints differ from standard multivariate distributions in that their expression is often more complicated, which makes their simulation delicate. In this chapter, two different methods have been considered for performing conditional simulations. Both are based on Monte Carlo techniques that deal with the numerical computation of integrals using stochastic approaches.

The first method is iterative. The underlying idea is to generate a sequence of states, the distributions of which converge to the target distribution. The implementation of these algorithms are often simplified by considering Markovian sequences. Prototypes of such algorithms are the Metropolis-Hastings algorithm and the Gibbs sampler.
The iterative algorithms are versatile. They can be applied to any distribution, and can accommodate various kinds of constraints. The main problem they raise is the establishment of their rate of convergence, that is, the determination of the number of iterations required to reach a distribution close to the target distribution up to a prespecified precision.

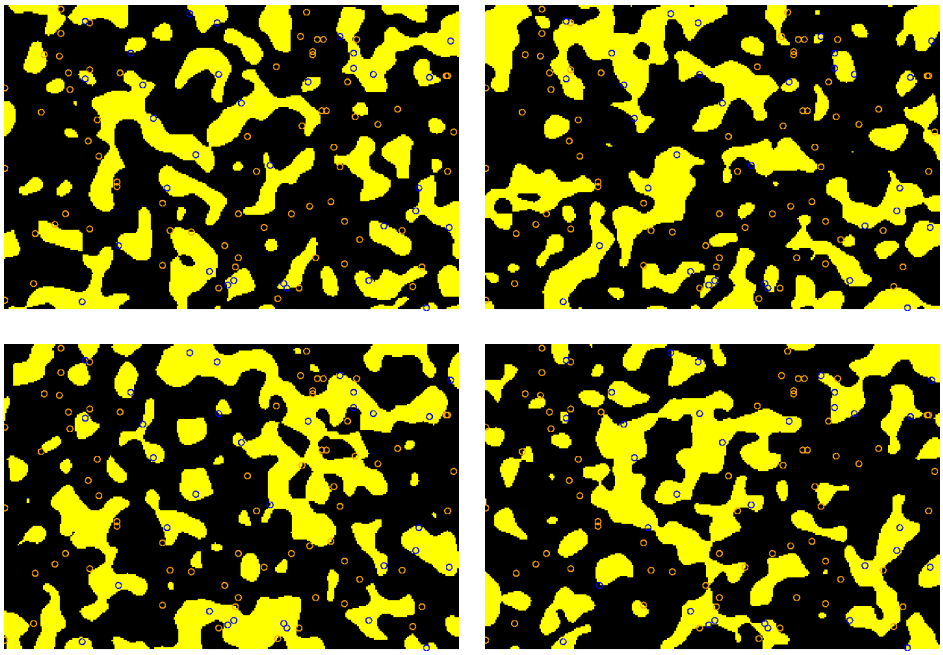


Figure 2.4: Four conditional simulations produced by particle filtering

The second method is sequential. It works in a finite number of steps equal to the number of components of the target multivariate distribution. At each step, a new component is simulated conditionally on the components already simulated and on the constraint prescribed on it. To compensate for the non integration of all constraints, several vectors of components (or particles in the particle filtering terminology) are simulated. Each particle is assigned a weight that reflects its compatibility with the constraints, and a resampling is performed to make all particles equiprobable. At the last step, one resampled particle is selected at random to serve as a conditional simulation. Thus the sequential method allows a progressive assimilation of the constraints. To a certain extent, the resampling makes the method Darwinian. Starting from a set of particles, only those that are the most compatible with the constraints (their environment) can survive.
The sequential method applies to distributions that can be written as a product of a standard multivariate distribution and functions of the components that specify the constraints. For that reason, the sequential methods are not so flexible as the iterative methods. The determination of the number of particles to simulate is also a pending issue.

In our experience, the sequential methods have turned out to be much faster than the iterative ones. Moreover, for the same running time, they provide more accurate simulations, in the sense that they better retrieve the statistical properties of the conditional model than the iterative methods. Of course, these are only provisional conclusions. A bit more will be said in the forthcoming chapters.

3

CONDITIONAL SIMULATION OF A BOOLEAN MODEL OF RESERVOIR USING PARTICLE FILTERING

Le schéma booléen est un prototype de modèle à base d'objets dans la mesure où il est la réunion d'objets de tailles, de formes et d'implantations indépendantes. Des réservoirs à deux faciès peuvent être modélisés, un faciès pouvant représenter des corps poreux (la "forme"), l'autre un facies imperméable (le "fond"). Un schéma Booléen conditionnel peut être simulé à partir d'un algorithme itératif sous des contraintes de données ponctuelles. Toutefois, ce type d'algorithme possède une vitesse de convergence souvent difficile à chiffrer. Ceci est une des raisons pour laquelle une alternative séquentielle est proposée. Celle-ci consiste à assimiler les points conditionnants de la forme un par un grâce à un filtrage particulaire. D'un autre côté, les points conditionnants du fond sont traités indépendamment par une procédure standard de rejet. En outre, une méthode pour quantifier la qualité des simulations conditionnelles a été développée. Au final, l'algorithme séquentiel surpasse la version itérative, du point de vue de la qualité des résultats et du temps de calcul.

3.1 Introduction

This chapter presents a first application of particle filtering to conditional simulations. The model considered is the Boolean model. This is a random set, the realizations of which are an aggregate on possibly overlapping compact subsets. It can be used to model two-facies reservoirs where one facies is composed of porous sandbodies, and the other facies is made of shales or clay that act as barriers to the circulation of fluids. The construction of the Boolean model rests on very simple, but not always very physically realistic rules. As the counterpart, this model possesses many interesting features. It is analytically tractable (Matheron, 1975), its parameters can be statistically inferred (Hall, 1988; Molchanov, 1997) and algorithms can be designed to simulate it, conditionally or not (Lantuéjoul, 2013). For that reason, the Boolean model has been used in many different fields, not only in oil industry (Haldorsen and MacDonald, 1987), but also in mining industry (Chautru, 1989), geology (Dubrule, 1989), metallurgy (Greco et al., 1979), material sciences (Jeulin, 2021) among others.

This chapter is organized as follows. Section 3.2 specifies the ingredients required for the construction of the Boolean model, as well as how to characterize them from a statistical point of view. It also makes explicit the definition of the Boolean model, reviews its main properties and gives an algorithm to simulate it non conditionally. Section 3.3 presents two algorithms for conditional simulation. The traditional one resting on Markovian iterations is compared to a novel one based on particle filtering. Their performances are compared from different points of view such as accuracy and precision, versatility, running time and memory footprint. Finally, section 3.4 draws conclusions while mentioning some pending questions.

3.2 Boolean model

3.2.1 Definitions

The intuitive idea that underlies the construction of the Boolean model consists of randomly implanting independent objects in space, and taking their union (see Matheron (1967)). Of course we must precisely specify what we mean by the objects and their implantation.

Objects. In what follows, each object is a nonempty random compact subset of $\mathbb{R}^d$. To specify its randomness, two different cases must be considered. If the object takes a simple shape, then this shape can be specified by a set of parameters and their joint distribution is required. For example, a rectangular object in the plane can be characterized by the 5-dimensional joint distributions of its center (two coordinates), its width, height, and orientation. For more complicated objects, this description is no longer possible. In that case, it is convenient to resort to the random set theory developed by Matheron (1975). The statistical properties of a random closed set A are equivalently characterized by its *hitting functional* or its *avoiding functional* that respectively assign each compact subset K the probability $T_A(K)$ that K intersects with A, or the probability $Q(K)$ that K is disjoint from A[1]

$$T_A(K) = P\{A \cap K \neq \varnothing\} \qquad K \in \mathcal{K}(\mathbb{R}^d)$$
$$Q_A(K) = P\{A \cap K = \varnothing\} \qquad K \in \mathcal{K}(\mathbb{R}^d)$$

with $\mathcal{K}(\mathbb{R}^d)$ the family of compact subsets of $\mathbb{R}^d$. From a practical viewpoint, the hitting and the avoiding functionals act to closed random sets exactly as distribution functions and complementary distribution functions to random variables.

Each object is also associated a reference point that serves as its implantation point; for instance, the center of a circular object, but this choice is by no means compulsory.

Implantation. The objects are implanted at the points of a Poisson process (Kingman, 1992). This particular point process depends on a measurable and positive function θ, defined on $\mathbb{R}^d$ and called *intensity function*. To shorten the notation we denote by $\theta(B)$ the integral of θ over the Borelian B. Of course, $0 \leq \theta(B) \leq \infty$.

A simple way to describe the spatial distribution of a Poisson point process is via its counting measure. This measure assigns to each Borelian of $\mathbb{R}^d$ the random number of points fallen within it. It satisfies the following two characteristic properties:

(i) the number of points n within a Borelian B satisfying $\theta(B) < \infty$ is *Poisson distributed* with mean $\theta(B)$

$$P_B(n) = \exp\big(-\theta(B)\big)\,\frac{\theta^n(B)}{n!}$$

(ii) the numbers of points within pairwise disjoint Borelians are *mutually independent.*

An important consequence of this definition is that if B contains n points, then these n points are independently located within B according to the same p.d.f.

$$f(x) = \frac{\theta(x)}{\theta(B)} \qquad x \in B.$$

Definition: A Boolean model is defined as a union of independent objects A_s located at the points s of a Poisson process:

$$X = \bigcup_{s \in \mathcal{P}} A_s, \tag{3.1}$$

[1]Strictly speaking, the hitting and the avoiding functionals of A are defined on the family $\mathcal{B}(\mathbb{R}^d)$ of Borel subsets. However, it has been shown that these functionals are completely specified by their restrictions to $\mathcal{K}(\mathbb{R}^d)$.

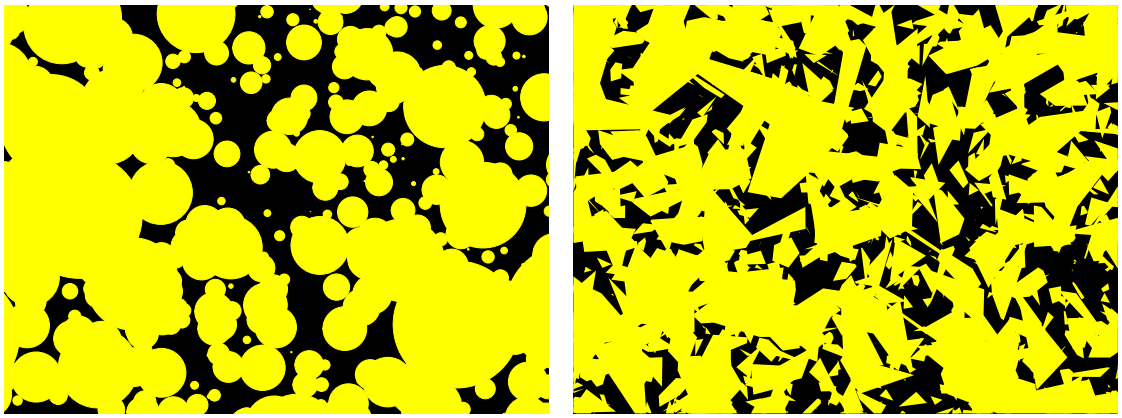


Figure 3.1: Two examples of Boolean models with a constant intensity function. (left) The typical object is a circle with a radius following an exponential distribution, and (right) the typical object is a Poisson polygon.

where $\mathcal{P}$ is a Poisson point process with intensity function θ, and $(A_s), s \in \mathbb{R}^d$ is a population of independent objects. The distribution of each object A_s is equivalently specified by its hitting functional $T_s(K) = P\{A_s \cap K \neq \emptyset\}$ or its avoiding functional $Q_s(K) = P\{A_s \cap K = \emptyset\}$.

A Boolean model is said to be *stationary* if the Poisson intensity is constant and if all objects have the same hitting or avoiding functionals up to a translation. Figure 3.1 shows a realization of two examples of stationary Boolean models, one with discs and the other with polygons. We give the parameters of the first model as it will be used in the rest of the chapter. The Poisson intensity is $\theta = 10$ and the disks have exponential radii with parameter $a = 7.22$, which warrants that the proportion of space occupied by the objects is 70%. The simulation domain is 8×6.

3.2.2 Properties

It can be shown by Lantuéjoul (2002) that the mean number of objects hitting the compact subset K is equal to

$$\vartheta(K) = \int_{\mathbb{R}^d} \theta(s) T_s(K)\, ds \tag{3.2}$$

This quantity may be finite or infinite. If $\vartheta(K) < \infty$ for any $K \in \mathcal{K}$, the Boolean model is said to be of *finite order*. In this chapter, only Boolean models of finite order are considered. This property implies not only that X is a random closed set, but also that the number of such objects hitting any compact subset K is Poisson distributed with mean $\vartheta(K)$. In particular, the avoiding functional of X is

$$Q_X(K) = \exp\{-\vartheta(K)\} \qquad K \in \mathcal{K} \tag{3.3}$$

From this formula, a number of stability properties can be derived, such as the union of two *independent* Boolean models is a Boolean model, and the intersection of a Boolean domain and a compact subset is a Boolean model.

Let us be a little more specific about this second property. Starting from a compact subset V of $\mathbb{R}^d$, the following two sets are defined

$$X^V = \bigcup_{\substack{s \in \mathcal{P} \\ A_s \cap V = \varnothing}} A_s \qquad \text{and} \qquad X_V = \bigcup_{\substack{s \in \mathcal{P} \\ A_s \cap V \neq \varnothing}} A_s. \tag{3.4}$$

Despite their cumbersome notations, these two random sets possess simple interpretations (cf. Fig. 3.2):

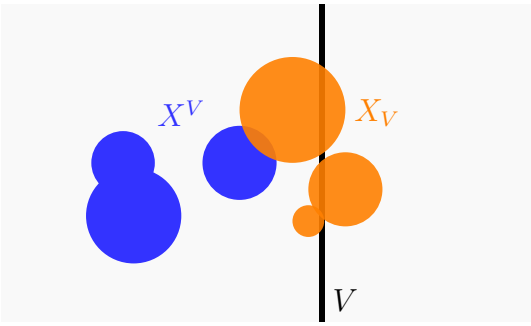


Figure 3.2: Decomposition of a Boolean model X into two random sets. X_V and X^V are the union of the objects respectively hitting and avoiding V.

Proposition 1 (Troncoso et al. (2022)). *$X = X_V \cup X^V$. Moreover, X^V and X_V are two independent Boolean models with the following avoiding functionals:*

$$Q_{X^V}(K) = \frac{Q_X(K \cup V)}{Q_X(V)} \qquad Q_{X_V}(K) = \frac{Q_X(K)\, Q_X(V)}{Q_X(K \cup V)}. \tag{3.5}$$

Fig. 3.3 shows the realizations of X_V and X^V corresponding to the left realization of Fig. 3.1. The points of V are represented by small black circles, and the objects of X are depicted as orange discs for X_V and blue discs for X^V. The decomposition of X into X^V and X_V reveals many hidden objects. Note also that the objects of X_V are much larger than those of X^V.

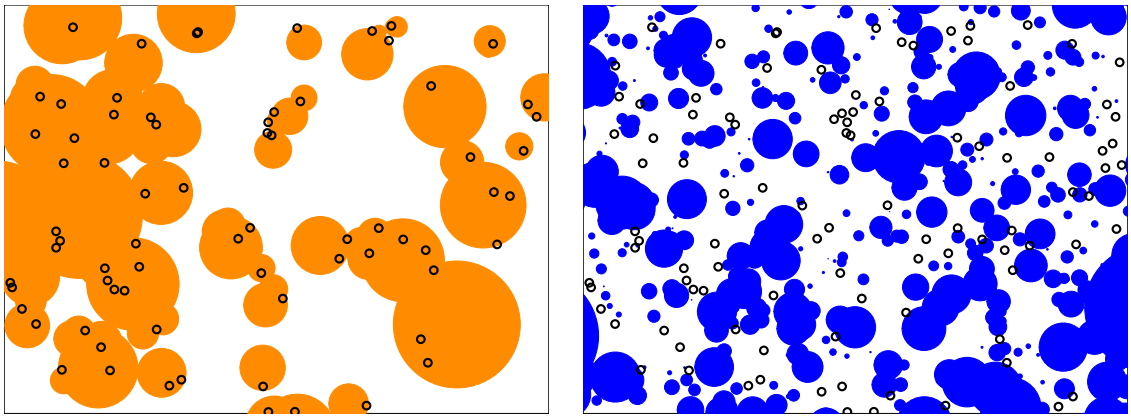


Figure 3.3: Taking V a set of points and a Boolean model X, an example of the decomposition into the sets X_V and X^V is shown. (left) A X_V type, and (right) a X^V type.

More generally, the construction of X^V and X_V can be iterated. Let U and V be two compact subsets of $\mathbb{R}^d$. Proposition 1 implies that the random set:

$$X_U^V = (X^V)_U = (X_U)^V$$

is a Boolean model with the avoiding functional:

$$Q_{X_U^V}(K) = \frac{Q_X(V \cup K)}{Q_X(V)} \frac{Q_X(U \cup V)}{Q_X(U \cup V \cup K)} = \frac{Q_{X^V}(K)}{Q_{X^{U \cup V}}(K)}. \tag{3.6}$$

3.2.3 Simulation

The objective of this section is to produce realizations of the Boolean model X in a compact domain D. According to the previous notations, this amounts to produce realizations of all objects of X_D. To simplify matters, the objects are assumed to contain their reference point.

Owing to the properties of the Poisson point process, this simulation problem is elementary when the objects are uniformly bounded, in the sense that all objects can be enclosed in a ball of fixed radius, say ρ. In that case, it suffices to generate a Poisson point process in the domain D dilated by a ball of radius ρ, replace each Poisson point by an object and take the union.

Things are not so simple when the objects are not uniformly bounded because remotely implanted objects can hit the simulation domain. To explain the algorithm, we shall mainly dwell on the case of a stationary Boolean model of discs.

Let θ be the intensity of the Poisson point process. Let also f be the p.d.f. (with possible unbounded support) of the disc radii. Since D is compact, there is no inconvenience in assuming that it is a disc centered at the origin (otherwise enclose D within such a disc, say D', perform simulations in D' and restrict them to D). The radius of D is denoted by r_D.
According to the stability properties, X_D is known to be a Boolean model. To identify its parameters, we remark that an object of radius R implanted at $x \in \mathbb{R}^2$ hits D if and only if $r_D + R > |x|$, which takes place with probability 1 if $x \in D$ and $\bar{F}(|x| - r_D)$ otherwise ($\bar{F}$ is the complementary distribution function of f). Accordingly, the Poisson intensity of X_D is equal to

$$\theta_D(x) = \begin{cases} \theta & x \in D \\ \theta\, \bar{F}(|x| - r_D) & x \in D^c \end{cases}$$

from which it follows that the mean number of objects of X_D is

$$\vartheta(D) = \theta \left(\pi r_D^2 + \int_{D^c} \bar{F}(|x| - r_D)\, dx \right).$$

Since X has been assumed to be of finite order, this quantity is finite (in particular f is of finite variance). Then X_D is made of a Poisson number of objects, and each object is independently located according the p.d.f. $\theta_D(x)/\vartheta(D)$. Moreover, the radius distribution of an object located at x has its p.d.f. equal to

$$f_D(r|x) = \frac{f(r)}{\bar{F}\left((|x| - r_D)_+ \right)}\, \mathbb{1}_{r > |x| - r_D}$$

with $u_+ = \sup(u, 0)$.

To complete these results, it is interesting to specify the p.d.f. f_D of the simulated disc radii:

$$f_D(r) = \int_{\mathbb{R}^2} \frac{\theta_D(x)}{\vartheta(D)}\, f(r|x)\, dx = \frac{\theta}{\vartheta(D)}\, f(r) \left(\pi r_D^2 + 2\pi r_D r + \pi r^2 \right)$$

Moreover, direct calculation gives

$$\vartheta(D) = \theta \left(\pi r_D^2 + 2\pi r_D\, m_1 + \pi m_2 \right) \tag{3.7}$$

where m_1 and m_2 are the first two moments of f. This shows that f_D can be written

$$f_D(r) = p_0\, g_0(r) + p_1\, g_1(r) + p_2\, g_2(r), \tag{3.8}$$

where the g_i's are weighted p.d.f. versions of f

$$g_i(r) = \frac{r^i f(r)}{m_i} \qquad i = 0, 1, 2$$

and the p_i's are the weights

$$p_0 = \frac{\pi r_D^2}{p} \qquad p_1 = \frac{2\pi r_D m_1}{p} \qquad p_2 = \frac{\pi m_2}{p}$$

with $p = \pi r_D^2 + 2\pi r_D\, m_1 + \pi m_2$.

Thus formula (3.8) provides a simple algorithm to simulate the objects of X_D. At first, the number of objects is generated from a Poisson distribution with mean $\vartheta(D)$. Then for each object, a radius r is sampled from f_D and the disc is implanted at a uniform point x within the disc of radius $r + r_D$ centered at the origin.

It should be mentioned that this algorithm applies even when D is reduced to a point (this remark will be useful later on). Moreover, it can be extended when the objects are nonconvex. For more details, see Lantuéjoul (2013).

3.3 Conditional Boolean model

3.3.1 Definition and properties

The Boolean model X splits $\mathbb{R}^d$ into two sets, namely the union of the objects, or *foreground*, and their complement, or *background*. Let also B and F be two disjoint compact subsets of $\mathbb{R}^d$. We are concerned with the realizations of X that contain F in their foreground and B in their background. The random set associated with these realizations is called the *conditional Boolean model* and denoted by $X(B, F)$.

By construction, $X(B, F)$ is a closed random set. If F is finite, then its avoiding functional can be derived using the inclusion-exclusion formula. We explicitly find

$$Q_{X(B,F)}(K) = \frac{\sum_{C\subset F}(-1)^{|C|} Q_X(K \cup B \cup C)}{\sum_{C\subset F}(-1)^{|C|} Q_X(B \cup C)}\ , \tag{3.9}$$

where $|C|$ denotes the cardinality of C. This equation shows that, in general, a conditional Boolean model is not a Boolean model.

If formula (3.9) is applied to the Boolean model X_F, and if Q_{X_F} is replaced by its expression (3.5), it appears that

$$Q_{X_F(B,F)}(K) = \frac{Q_X(B \cup F)}{Q_X(K \cup B \cup F)} \frac{\sum_{C\subset F}(-1)^{|C|} Q_X(K \cup B \cup C)}{\sum_{C\subset F}(-1)^{|C|} Q_X(B \cup C)}.$$

Thus, $Q_{X_F(B,F)}(K)$ can be written as the product of two fractions. According to formula (3.9), the first fraction is the reciprocal of $Q_{X(B\cup F,\varnothing)}(K)$, and the second fraction is nothing but $Q_{X(B,F)}(K)$. Since the objects of $X(B \cup F, \varnothing)$ and $X(B, F)$ are necessarily different, these two random sets are independent. It follows

Proposition 2 (Decomposition property)**.** *: $X(B, F)$ is equivalent in distribution as the union of the two independent Boolean models $X(B \cup F, \varnothing) \equiv X^{B\cup F}$ and $X_F(B, F)$.*

$$X(B, F) \equiv X(B \cup F, \varnothing) \cup X_F(B, F) \tag{3.10}$$

3.3.2 Conditional simulation: iterative method

The objective of this section is to present the algorithm to perform a conditional simulation of a Boolean model using a Markov Chain Monte Carlo algorithm coupled with a restriction technique (Gedler, 1991; Lantuéjoul, 2002; Allard et al., 2006). The Boolean model considered is the one presented in the non-conditional section: Poisson intensity θ and circular objects with radius p.d.f. f. This model is subject to the foreground condition F and the background condition B. The simulation domain D is the disc of radius r_D centered at the origin. To avoid tedious notations, B and F are supposed to be part of D.

We already know that in the unconditional model the number of objects hitting D follows a Poisson distribution with mean $\vartheta(D)$ (Lantuéjoul, 2002). This distribution can be simulated as the limit distribution of a birth-and-death process using Metropolis algorithm. Suppose that n objects are present at a given step. At the next step, three situations can occur:

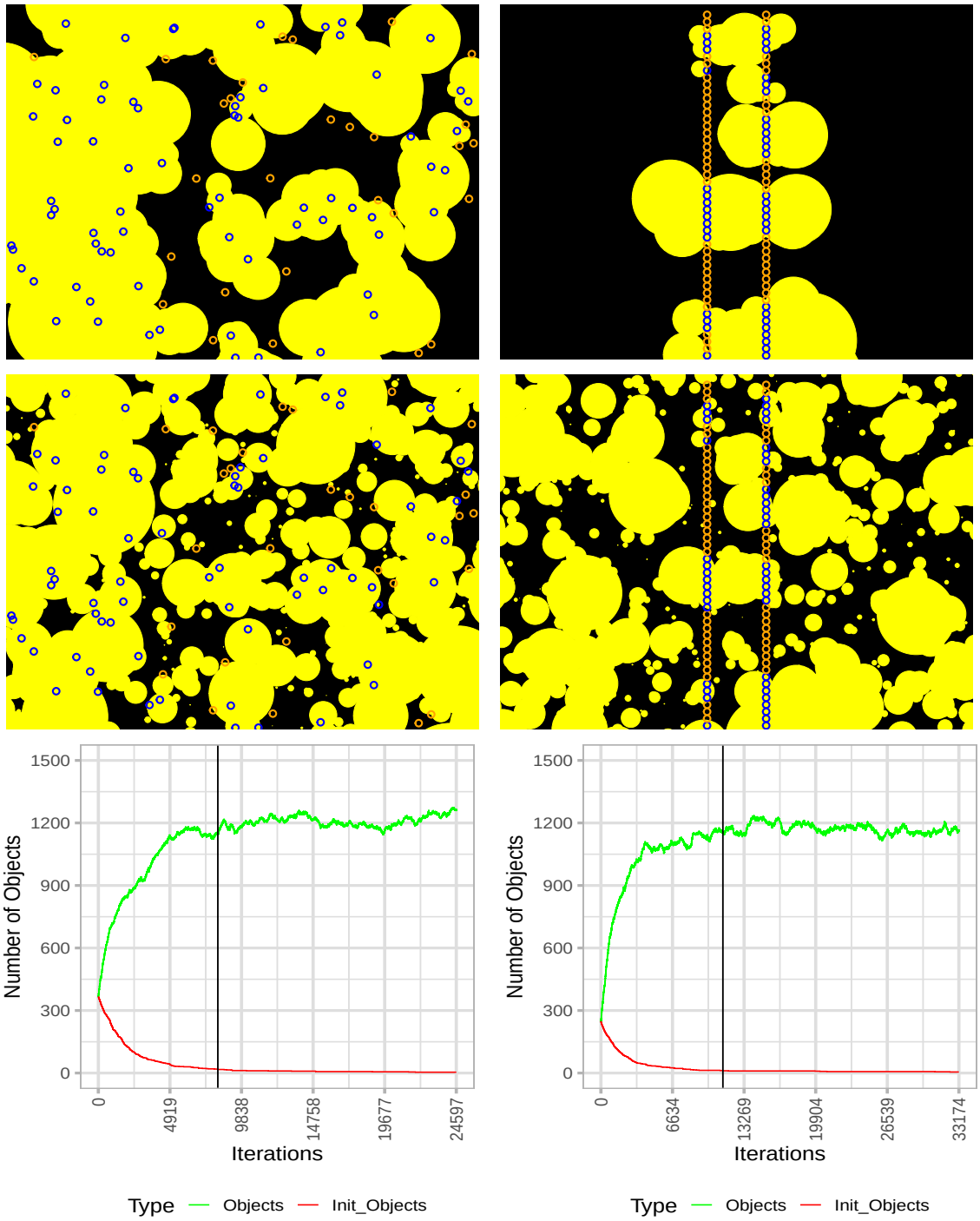


Figure 3.4: Realizations of $X(B, F)$ produced by the iterative method using two configurations of point constraints. From top to bottom, the initial populations of objects, the simulations, and the graphs showing the total number of objects (green line) and the remaining number of initial objects (red line) versus the iteration order. The vertical black lines indicate the iteration orders at which no more than 5% of the initial objects are still present.

- With probability $P(n, n+1) = \vartheta(D)/(\vartheta(D) + n + 1)$, a new object hitting D is generated and inserted to the population;
- With probability $P(n, n-1) = n/(\vartheta(D)+n)$, an object present in the population is selected and removed;
- With probability $P(n, n) = 1 - P(n, n+1) - P(n, n-1)$, the population is left unchanged.

This iterative algorithm is made conditional by prescribing all conditions to be satisfied at each

iteration.
More precisely, an initial population of objects satisfying all conditions is first generated. This can possibly be done by sequentially generating objects hitting D. All objects that are overlapping with B are systematically discarded. The same happens to objects that are not the first ones to cover points of F. This process is continued until all points of F are covered by an object.

Then at each iteration, any change proposed by the unconditional algorithm is performed as long as conditions are not disrupted. If a new object has been generated, this object is effectively inserted to the population, provided that it does not overlap with B. Similarly, if an old object is to be discarded, it is effectively removed provided that it is not the only one to cover points of F.

Figure 3.4 shows simulations produced by this iterative algorithm using two different configurations (purely random and fully structured) of 100 conditioning data points. In both cases, the algorithm has been stopped at iteration 3τ, where τ is the iteration order at which only 5% of the initial objects still remain present.

Just like any iterative algorithm, this one suffers from two limitations, namely the existence of a burn-in phase and of periods of metastable regime that make the establishment of the rate of convergence delicate. This is the reason why an empirical rule of thumb of 3τ has been retained. Additionally, formula (3.10) suggests that it would be interesting to limit the use of the iterative algorithm to simulate only $X_F(B, F)$. However, this is not possible because the value of $\vartheta(F)$ is unknown.

To overcome the limitations of this iterative algorithm, a sequential algorithm is proposed.

3.3.3 Conditional simulation: sequential method

The section presents the algorithm to perform a conditional simulation of a Boolean model using particle filtering. This algorithm has been recently published in a slightly different presentation (Troncoso et al., 2022). It exploits the dichotomy between the avoiding model $X(B \cup F, \varnothing)$ and the hitting model $X_F(B, F)$ provided by the decomposition property (3.10). The model, the simulation domain and the conditions are the same as in the previous section.

The simulation of the avoiding model is straightforward. It suffices to perform a non-conditional simulation of the Boolean model in D and discard the objects containing conditioning data points.

The simulation of the hitting model, on the other hand, is more subtle. It rests on a sequential assimilation of the conditioning points of the foreground, which requires them to be ordered (all total orders are acceptable). Suppose that F consists of k points, say $F = \{c_1, ..., c_k\}$, or $F = c_{\leq k}$ for short. Then we write $F_i = \{c_{\leq i}\}$ for each $i = 1, ..., k$. We also put $F_0 = \varnothing$ by convention.
For each $i = 1, ..., k$, let N_i be the number of objects that contain c_i and avoid $B \cup F_{i-1}$. By definition, the union of these objects coincide with the set $X_{\{c_i\}}^{B \cup F_{i-1}}$ that has been shown to be a Boolean model. Accordingly, N_i is Poisson distributed. Moreover, each object of X_F^B is counted exactly once, which stems that the N_i's are mutually independent.

In what follows, the N_i's are going to be conditionally simulated using particle filtering (cf. algorithm 4). To achieve this goal, the target distribution considered has for expression

$$f(n_{\leq k}) = \alpha_k \prod_{i \leq k} p_i(n_i) \prod_{i \leq k} \mathbb{1}_{n(c_i) \geq 1}$$

where p_i is the Poisson distribution of N_i, $n(c_i)$ stands for the number of objects covering c_i and avoiding B, and α_k is a normalization factor. Regarding the auxiliary distributions used to approximate f, they are abbreviated versions of f

$$f_i(x_{\leq i}) = \alpha_i \prod_{i' \leq i} p_{i'}(n_{i'}) \prod_{i' \leq i} h_{i'}(x_{i'}) \qquad i = 1, ..., k$$

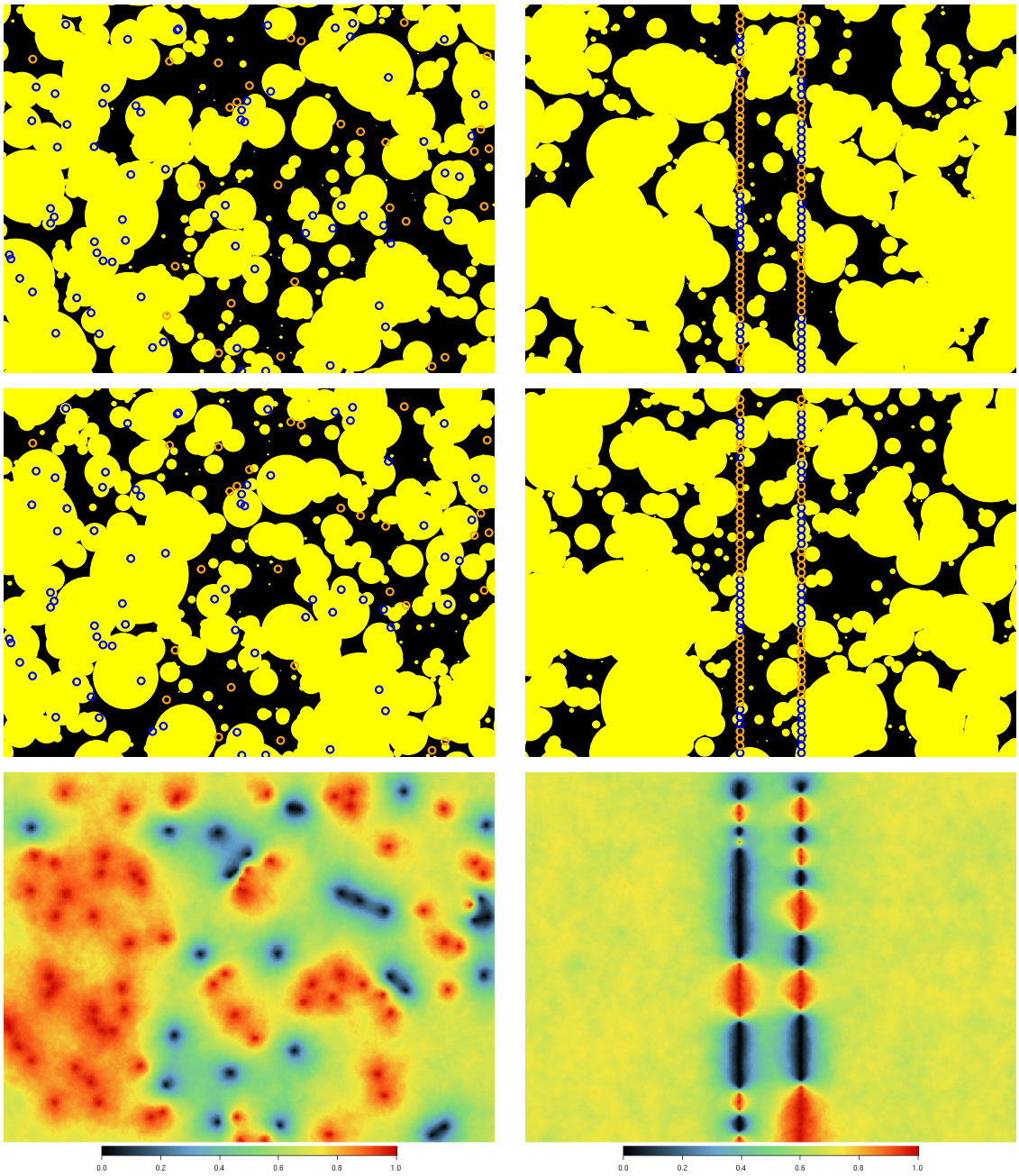


Figure 3.5: Realizations of $X(B, F)$ produced by the sequential method using two configurations of data points. Top and middle, two conditional simulations, and bottom the probability map estimated from 500 independent simulations.

Finally, the importance sampling distributions are nothing but the aforementioned Poisson distributions

$$g_i(n_i|n_{<i}) = p_i(n_i) \qquad i = 1, ..., k.$$

from which it results that the weight assigned to the particle $(n_{1:i})$ is:

$$w_i = \frac{f_i(n_{\leq i})}{f_{i-1}(n_{<i})\, g_i(n_i|n_{<i})} = \frac{\alpha_i}{\alpha_{i-1}} \mathbb{1}_{n(c_i)\geq 1}.$$

It should be mentioned that the proportionality coefficient α_i/α_{i-1} does not depend on the particles. It does not depend either on the Poisson intensities, which is rather fortunate because they are usually unknown. Note also that these Poisson intensities are not even required to generate the n_i's. To simulate each n_i, it suffices to perform a non-conditional simulation of the

Boolean model $X_{\{c_i\}}$ and discard the objects hitting $B \cup F_{i-1}$. This rejection procedure yields not only n_i but also the objects of $X_{\{c_i\}}^{B \cup F_{i-1}}$.

Fig. 3.5 shows two realizations produced by the sequential algorithm, using the same conditioning data points as in Fig. 3.4. 200 particles have been used to assimilate the points of F. The average of 500 independent conditional simulations has also be computed to estimate the probability that each point of the simulation domain is covered by objects (see bottom of Fig. 3.5).

3.3.4 Evaluation of the simulation algorithms

An iterative algorithm and a sequential algorithm have been presented to perform conditional simulations of a Boolean model subject to point observations. The natural question that arises is how do they compare from the points of view of the quality of results, the running time and the memory footprint.

In principle, this evaluation can be made by comparing the empirical statistical properties derived from a set of conditional simulations and the theoretical properties of the conditional Boolean model that are available from its avoiding functional. However, this approach is purely formal because the model is non-stationary and cannot provide anything else but local comparisons, and also because the calculation of this avoiding functional becomes intractable as soon as the number of foreground conditioning data points exceeds 5 or 6. Another approach is therefore necessary.

The following approach recommended by X. Freulon rests on the idea that if the conditional simulation algorithm is correct and has been correctly implemented, then a conditional simulation subject to data values jointly drawn from a non-conditional simulation must be itself a non-conditional simulation (Freulon and de Fouquet, 1993). As a consequence, if 500 conditional simulations are performed this way, each with values at conditioning points randomized according that protocol, the statistical fluctuations that we should observe should be identical to those provided by 500 non-conditional simulations. This is what is to be tested.

The statistics used for comparison in this evaluation exercise consist of

- The distribution of the disc radii;
- The probabilities p and $q = 1-p$ that a point belongs to the foreground or the background;
- The variogram $\gamma_X(h) = P\{x \in X, x + h \notin X\}$.

It should be mentioned that the distribution of the radii that should be obtained is the mixture of 3 Gamma distributions presented in (3.8) because the Boolean simulations are performed on the domain D. Moreover, the two latter statistics can be written like

$$q = e^{-\theta\, K_A(o)}$$
$$\gamma_X(h) = q\left(1 - e^{-\theta\left[K_A(o) - K_A(h)\right]}\right)$$

where K_A is the *geometric covariogram* of the Boolean objects. This covariogram is defined as the expected area of an object and its translates

$$K_A(h) = \mathbb{E}\{|A \cap \tau_h A|\} \qquad h \in \mathbb{R}^2$$

For discs with exponential radii, this covariogram satisfies $K_A(o) = 2\pi/a^2$, as well as

$$K_A(o) - K_A(h) = \frac{\pi|h|}{a}\left[K_0\left(\frac{a|h|}{2}\right)L_1\left(\frac{a|h|}{2}\right) + K_1\left(\frac{a|h|}{2}\right)L_0\left(\frac{a|h|}{2}\right)\right]$$

where K_0 and K_1 are modified Bessel functions, and L_0 and L_1 are modified Struve functions (Abramovitz and Stegun, 1970). For a proof, cf. Appendix A.1).

At first, Fig. 3.6 shows the statistical fluctuations stemming from 500 non-conditional simulations performed in a 8×6 domain. Regarding the point statistics, we should have $p = 0.7$. Since p is estimated using $n = 500$ simulations, its estimator $\hat{p}$ should lie within the interval $[p - 2\sqrt{pq/n}, p + 2\sqrt{pq/n}] \approx [0.66, 0.74]$ with 95% chance. The estimates below and above these interval limits are respectively represented in yellow and red.

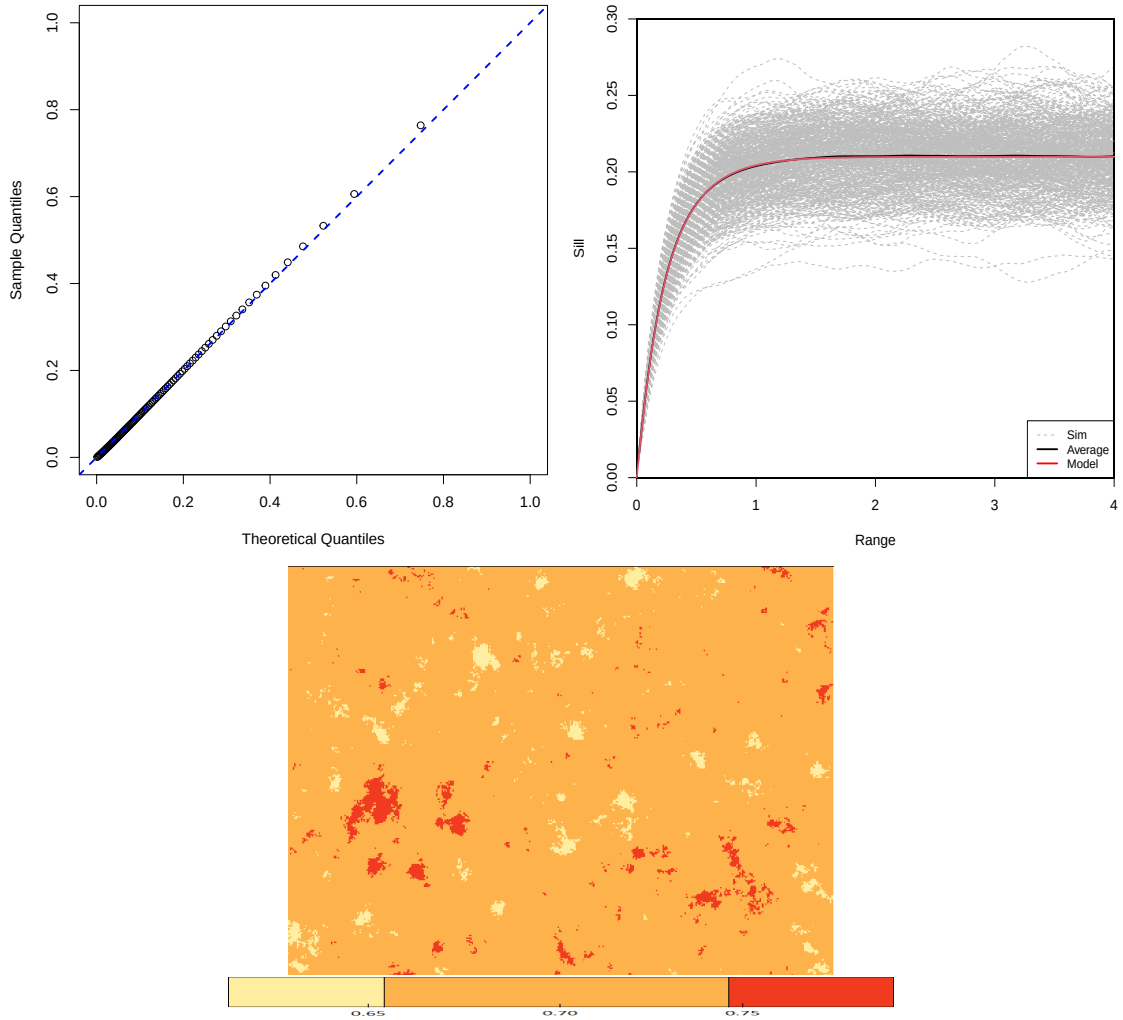


Figure 3.6: Result of 500 non-conditional simulations of X. Top left the QQ-plot of the disc radii, top right the variogram of each simulation with its average and model, and bottom the probability map with a threshold of two standard deviations.

Then 500 conditional simulations have been performed using each algorithm (iterative and sequential) and for each geographical configuration of data points (purely random and fully structured) in the same simulation domain (8×6). The results are given in Fig. 3.7 for the iterative algorithm and in Fig. 3.8 for the sequential algorithm.

It appears that the QQ-plots shows that the experimental radius distributions are well-fitted by the mixture of Gamma distributions. Note however that the QQ-plots are slightly more distorted for iterative simulations than for sequential simulations, but the amplitude of the distortions is really small.

The situation is not the same for the variograms. There is clear evidence that the sequential algorithm reproduces the variogram model much better than the iterative algorithm. The iterative simulations yield experimental variograms that dramatically overestimate the variogram model. Regarding the sequential simulations, an overestimation can also be observed in the case of a configuration of purely random data points, but it is quite small. The fit is almost perfect for a fully structured configuration of data points.

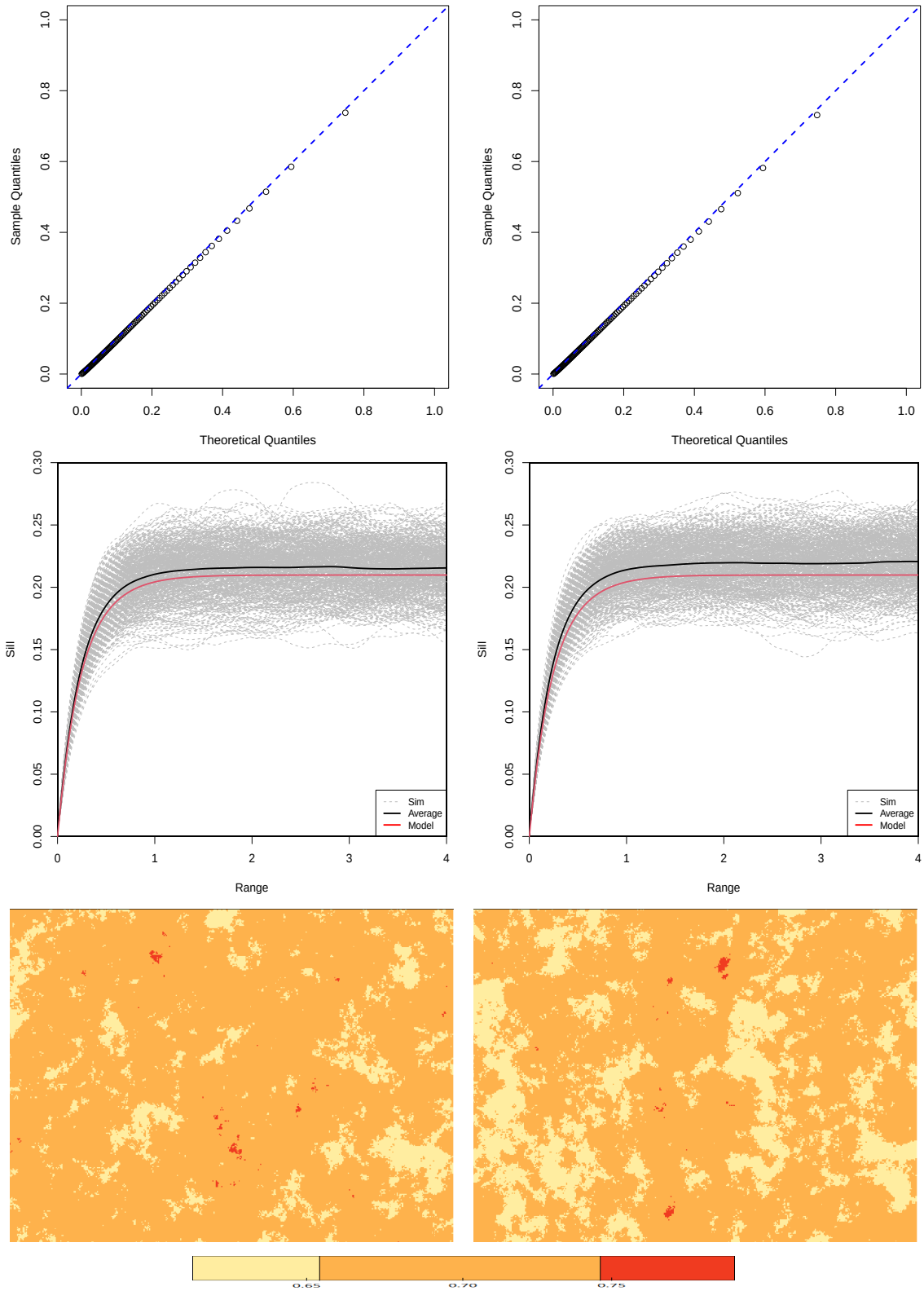


Figure 3.7: Result from 500 conditional simulations using the iterative algorithm and randomizing the conditioning points values. On the left, purely random data points, and on the right the fully structured data points. From top to bottom, the QQ-plot of the disc radii, the variogram of each simulation with its average and model, and the probability map with a threshold of two standard deviations.

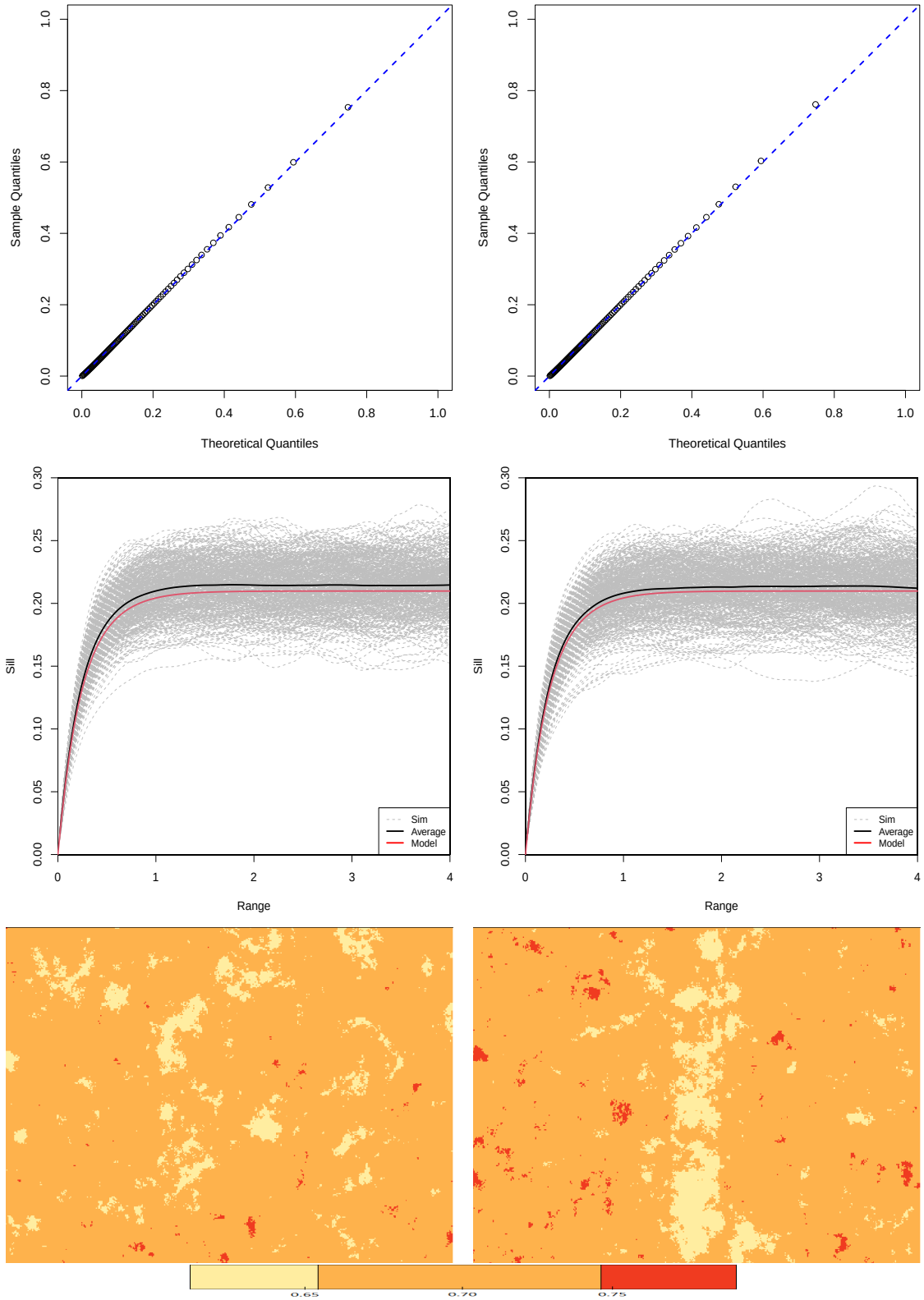


Figure 3.8: Result from 500 conditional simulations using the sequential algorithm (200 particles) and randomizing the conditioning points values. On the left, purely random data points, and on the right the fully structured data points. From top to bottom, the QQ-plot of the disc radii, the variogram of each simulation with its average and model, and the probability map with a threshold of two standard deviations.

The probability maps derived from the point statistics are not satisfying for the iterative simulations. There are too many points lying outside the 95% interval. They are also organized in too large components (especially for fully structured configuration of points), and an important asymmetry can be observed towards small values. Things clearly improve with the sequential simulations, notably in terms of point proportions and symmetry, but artifacts are still present (see for example the bottom right of Fig. 3.8).

It finally appears that the comparison results are not as satisfactory as one could expect. To explain this, one could think that the number of iterations or the number of particles have not been adequately chosen and should be reconsidered[2]. However, another argument may be put forward. Starting from the fact that 100 conditioning data points give rise to 2^{100} configurations of values, one can wonder whether the 500 configurations used to produce the 500 conditional simulations are really sufficient to represent the whole set of configurations. Because of the size and the shape of the objects, all configurations are not equiprobable. Certain configurations are largely privileged, whereas some others have very little chance to occur. Using 500 configurations, one cannot guarantee that the configurations are represented with their correct weights.
In an attempt to alleviate the influence of the objects, it has been considered to randomize not only the values but also the locations of the conditioning data points. In the following exercise, each simulation (iterative or sequential) is performed subject to 100 conditioning points independently and uniformly scattered throughout the simulation domain. The results are shown in Fig. 3.9. Clearly, an improvement is observed, especially regarding the probability maps. It can also be observed that the sequential algorithm yields much better statistics than the iterative one for the three statistics considered.

Of course, other experiments should be conducted before drawing definitive conclusions, but this exercise clearly does not invalidate the representativity argument that has been raised.

3.4 Conclusions

A sequential algorithm based on particle filtering has been designed to simulate a Boolean model subject to point conditions. This algorithm is versatile as it can be applied to any workspace of any dimension. It can also handle non stationary situations and can manage objects of any shape and size. Moreover, it is easy to implement. Compared to the current iterative algorithm, the sequential algorithm is rather fast as it can target the conditioning effort on the random objects hitting the foreground conditions. It undoubtedly requires more memory storage but its amount is far from prohibitive. In the example considered (objects are discs with exponential radii), 200 particles are sufficient to produce accurate simulations. In particular, their statistical fluctuations, observed on 500 simulations, are in close conformity with those of the model on configurations of 100 either purely random or fully structured data points.

Of course, any novel algorithm cannot be designed without raising a number of pending questions. At the present stage, there is still uncertainty concerning the number of particles required to efficiently run the sequential algorithm. One does not know whether the algorithm can be generalized to accommodate regional constraints such as connectivity constraints. Finally, the procedure developed to validate the correct implementation of the algorithm via the statistical fluctuations produced needs further developments.

[2] As a matter of fact, sequential conditional simulations performed using 400, 800, 1600 and 3200 particles do not really show significant improvements. cf. Appendix A.2

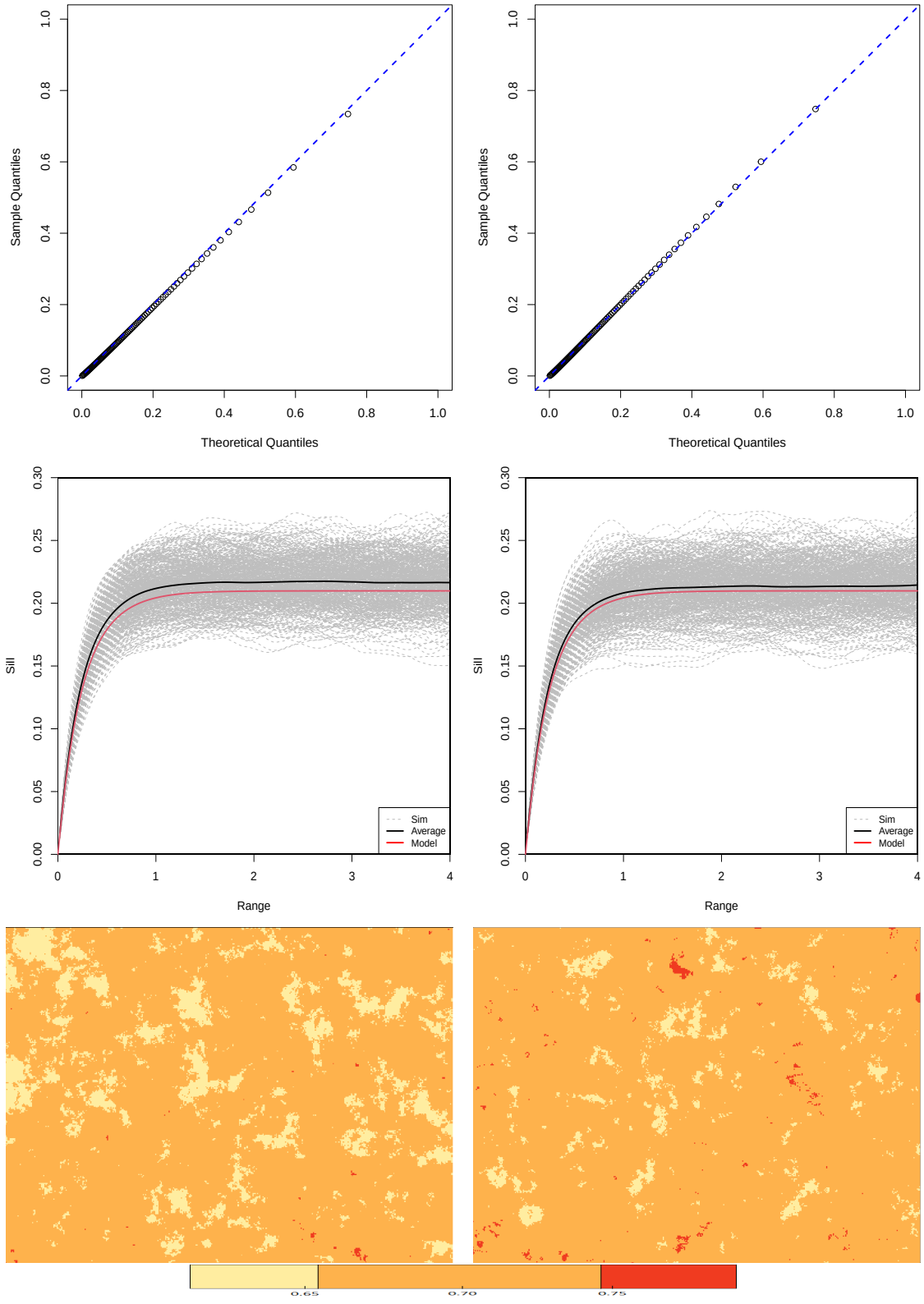


Figure 3.9: Result from 500 conditional simulations using the iterative algorithm (left) and the sequential algorithm (right), and randomizing both the locations and the values of the conditioning data points. From top to bottom, the QQ-plot of the disc radii, the variogram of each simulation with its average and model, and the probability map with a threshold of two standard deviations.

4

CONDITIONAL SIMULATION OF A PROCESS-BASED RESERVOIR MODEL USING PARTICLE FILTERING

Flumy est un modèle stochastique de réservoir reproduisant les processus sédimentaires de systèmes chenalisés méandriformes en environnement fluvial ou turbiditique. Les données conditionnantes peuvent inclure des données de faciès le long de puits, des cartes sismiques et des surfaces stratigraphiques. Actuellement, Flumy possède une méthode de simulation conditionnelle à des données de puits. Celle-ci altère les processus de manière itérative, modifiant au fil du temps le trajet du chenal pour assimiler l'information conditionnante des puits. Le développement d'une telle méthode de simulation conditionnelle requiert une compréhension profonde des processus, pour les modifier sans créer d'artefacts ou de biais indésirables dans le résultat. Pour éviter ces problèmes, une alternative séquentielle de simulation conditionnelle est proposée. Celle-ci met à profit la capacité de Flumy à créer des simulations non-conditionnelles de haute qualité en terme de reproduction d'hétérogénéité et de connexité. La méthode construit des tranches horizontales successives pendant qu'elle assimile leurs données de faciès grâce à un filtrage particulaire. Autre avantage, cette méthode permet l'assimilation d'autres types de données conditionnantes que les faciès aux puits et, en plus, s'implémente facilement, car les processus sédimentaires ne sont pas modifiés. Pour faire la comparaison entre les deux méthodes de simulation conditionnelle, itérative et séquentielle, deux cas sont étudiés. D'un côté, un cas synthétique, qui permet d'identifier les meilleures valeurs de paramètres pour le filtrage particulaire à partir d'une analyse de sensibilité. Les résultats semblent prometteurs, la méthode séquentielle donnant une meilleure reproduction des faciès aux puits que la méthode itérative actuelle. D'un autre côté, un cas réel (bassin de Loranca, en Espagne) où les résultats de la méthode séquentielle sont moins bons, probablement du fait que le nombre de particules a dû rester faible à cause de restrictions informatiques.

4.1 Introduction

This chapter presents a second application of particle filtering to conditional simulations. The objective is the simulation of reservoirs formed by meandering systems. Simulating deposits associated with meandering channels with statistical methods is a challenge due to the complex forms of the sedimentary bodies (point bar deposits, crevasse splays, oxbow lakes, etc) (Beucher et al., 2012).

That is the reason why Flumy, a process-based model developed at MINES ParisTech, is chosen, since it can accurately simulate the sedimentary bodies and their internal structures. Flumy was first dedicated exclusively to fluvial environments (Lopez, 2003; Lopez et al., 2009), but it has been now extended to turbiditic environments (Lemay, 2018). The simulations produced by

Flumy are reservoirs with high quality in terms of heterogeneity reproduction that permits to fully assess the connectivity of reservoir units.

Currently, there exists a conditioning method for Flumy, which considers different types of field data as constraints: well facies, stratigraphy maps or seismic data. However, some limitations exists in the produced reservoirs, such as possible incongruities between the grain sizes and the facies deposited at wells or distortions of sand body shapes.

In order to cope with these issues, an alternative to the conditioning method based on the *particle filtering* is proposed. This new method sequentially assimilates conditioning data while the reservoir is progressively built from bottom to top.

This chapter is organized as follows. Section 4.2 describes the sedimentary processes at work during the reservoir formation and how these processes are simulated by the Flumy model. Section 4.3 presents two methods for performing conditional simulations of the Flumy model. The traditional one, called the *dynamic conditioning method*, adapts the processes dynamically in order to adjust the meandering channel position toward the wells. The novel one, called the *sequential conditioning method*, is based on particle filtering. Their results and performances are compared from different points of view, such as well facies match, quality in reservoir produced, well grain size match and memory footprint using a synthetic case. Section 4.4 describes a real application case in Loranca Basin, Spain. Both conditioning methods are applied, and their performances are also compared. Finally, section 4.5 draws conclusions and perspectives.

4.2 Flumy Model

The Flumy model combines two approaches: stochastic, and process-based. The model reproduces the evolution of the meandering system and records in time the associated sedimentological deposits at the scale of the reservoir. The model simulates three main processes: (i) (horizontal) channel migration, (ii) aggradation, and (iii) avulsions. In the next section, these three main geological processes are described.

4.2.1 The meandering systems in a nutshell

In meandering systems, the sandy deposition is normally restricted to the main channel, or to partially or completely abandoned meandering loops; deposition of fines particles (silt and clay) occurs on levees and in flood basins (Walker, 1976).

Many important studies for these systems were performed from the middle of the XX^{th} century (Leopold and Wolman, 1957, 1960) based, generally, on geological studies (Marbut, 1896; Davis, 1903) and experiments (Friedkin, 1945). Nowadays, it is still difficult to experimentally reproduce a sustained, dynamic meandering channel pattern at laboratory scale. That is why the exact conditions for meandering systems depositions remain unclear (van Dijk et al., 2013). Some examples of studies for meandering systems are the river Klaralven in Norway and Sweden (Åke Sundborg, 1956), the Red River in Louisiana, USA (Harms et al., 1963), and the Wabash River in Illinois and Indiana, USA (Jackson II, 1976).

A high sinuosity characterizes meandering systems. Hence, as the meander bends, it migrates laterally and increases the bend amplitude while remaining single-threaded (van Dijk et al., 2013). This sinuosity comes from the erosion and deposition processes that occur while the channel transport sediments with different grain sizes.

Because of the meandering form, the channel has different local flow velocities depending on the spot of the measurement. Fig. 4.1 shows a representation of two positions in a meandering channel. At the apex of the meander (section B), the local flow velocity is higher at the outer bank (outer part of the channel, sometimes called cut bank), whereas it is lower at the inner bank (inner part of the channel). In the zone where there is no sinuosity (section A), the flow velocity

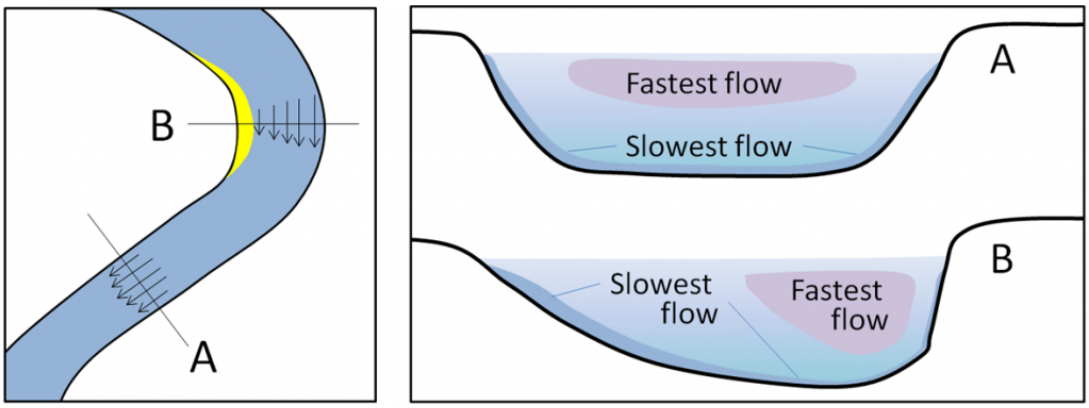


Figure 4.1: Primary flow velocities depending on the location. On the left, plan-views, on the right sections (figure taken from Earle (2016)).

is maximal at the top and decreases downward (more details in flow velocities in Pradhan et al. (2018)).

In the inner bank, since the flow velocity is lower than the outer bank, sandy *point bar* are deposited. The point bar is the main depositional environment, which builds laterally and downstream across the flood plain. Above the lag of the channel floor, sand is transported through the system as bedload.

Conversely, the erosion is performed at the outer bank, since the channel velocity in this area is higher. First, banks are undercut by fluvial erosion at the base and lower portion of the banks; second, bank retreat occurs by mass failure of the bank. Then, the bank sediment settles at the bank toe and armoring protects the bank against fast erosion and shifts the locus of the high flow velocity, which reduces the shear stress acting on the bank (van Dijk et al., 2013).

During overbank floods, the area that is submerged corresponds to the floodplain. It is flooded regularly and fine grain size sediments accumulate. This is the *aggradation* process. As a result, the proximal overbank flow deposits (sand to silt) build levees (elongated sandy to silty bodies that lie along the channel), and distal deposits (silt to clay) cover large areas in the floodplain. This results in a transition between the channel and the floodplain marked by a system of levees. The sedimentation rates of the floodplain are low and decrease with the distance to the channel due to the loss of velocity of the flow, which is no longer channelized. It results in the deposition of the fine-grained sediments away from the channel, leaving thin layers of silty to shaly alluvium. More details can be found in Bubnova (2018).

Levee breaches may occur during the evolution of the meandering channel. Suddenly, the bank can be fractured, generally, in the apex of the meandering channel, where the local velocity is higher. On one hand, the levee breach may cause diversion of part of the flow into the floodplain, with deposition of *crevasse splays*. The bank can be healed, and the meandering channel can continue into its original path. However, if the bank cannot be healed, the *avulsion* of the channels occurs, creating a new path. This process implies the reorganization of the hydrographic network, with changes in the depositional system of the floodplain and the possible construction of wetlands in the lower part of the topography.

During the development of the meandering system, a channel can be abandoned. The *channel abandonment* can be produced by an avulsion process or by cutoffs. Two cutoff processes are commonly recognized (Lewis and Lewin, 1983): *chute cutoff* and *neck cutoff*, both represented in Fig. 4.2.

On one hand, *chute cutoff* occurs when floods incise a floodplain channel, or chute, that evolves into the dominant conveyor of river discharge by the avulsion of the channel (Gay et al., 1998; Hooke, 1995). On the other hand, neck cutoff are produced during migration when the extremities of a meander get closer due to their outer bank erosion. When they encounter, the

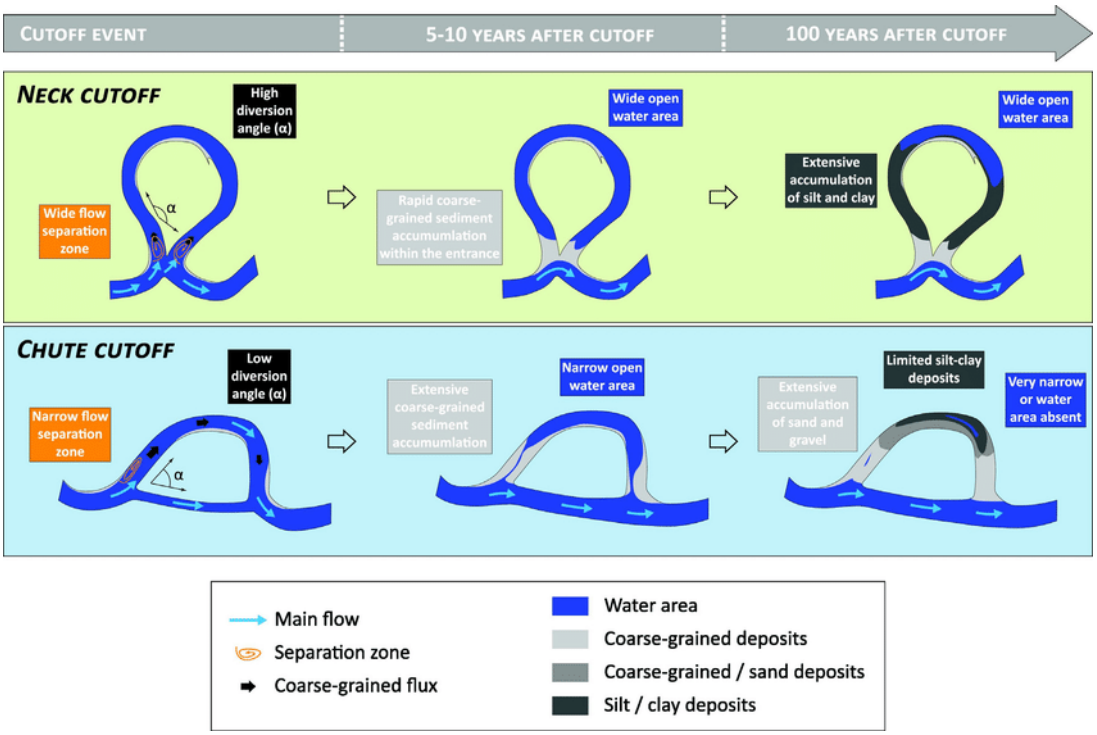


Figure 4.2: On top, the neck cutoff occurring during meander migration. At the bottom, the chute cutoff occurring during avulsions. In neck cutoff, a sand plug is quickly deposited at the intersection. Then, layers of fine sediments start to fill the oxbow lake. In chute cutoff, the abandonment is more gradual. Coarse-grained material is deposited first. Then, some sandy facies is deposited farther from the separation zone. Finally, fine grain size sediments are deposited in the farthest parts of the abandoned portion. (Figure taken from Dépret et al. (2017))

channel flows in a new path through the intersection. The abandoned channel is called *oxbow lake*.

4.2.2 Description of the model

Flumy can reproduce the geological phenomena previously described by simulating in time the three main processes: migration, aggradation and avulsion. A large set of different facies are deposited, allowing Flumy to generate heterogeneous reservoir models.

4.2.2.1 3D block model

The reservoir simulation is performed progressively. Time is discretized into time steps, called *iterations*. In Flumy, the material deposited by the different simulated processes can be classified into 13 categories or facies. This fine description can be further simplified in regrouping the initial facies into groups of *similar facies*. The number of groups retained is noted later as Nb. Nb can take the following values : two, three, seven, ten or thirteen. For example $Nb = 2$, corresponds to a description of the reservoir as a binary 3D image with sandy and permeable facies in one group and shaly and impermeable facies in the other group. The facies and their classification are listed in Fig. 4.3. Channel Fill and Pelagic facies are used for turbidite systems and are not deposited in fluvial environments.

The output of a Flumy simulation is a three-dimensional block stored in a *pillar grid*. Therefore, each $2D$ domain cell contains a stack of deposition units. Each deposition unit is informed with its thickness, its facies identifier, its grain size class and its age (iteration of deposition). Consequently, pillars have various thicknesses, which results in a block model with a non-constant topography.

Nb groups	PL	DR	WL	CF	MP	OB	LV	CS2	CCH	CS1	SP	PB	CL
2	OB									PB			
3	OB						LV			PB			
7	PL	DR	WL	OB			LV			PB			CL
10	PL	DR	WL	MP		OB	LV		CCH	CS1	PB		CL
13	PL	DR	WL	CF	MP	OB	LV	CS2	CCH	CS1	SP	PB	CL

Lithofacies			Grain
Id	Abr.	Full name	ϕ
0	UDF	Undefined	-
1	CL	Channel Lag	-8:-1
2	PB	Point Bar / LAPs	-1:3
3	SP	Sand Plug	0:3
4	CSI	Crevasse Splay I	0:3
5	CCh	Crev. Splay II Chan.	1:4
6	CSII	Crevasse Splay II	3:6
7	LV	Levee	3:6
8	OB	Overbank	7:14
9	MP	Mud Plug (flu)	11:12
10	CF	Channel Fill (tur)	11:12
11	WL	Wetland (flu)	13:14
12	DR	Draping (flu)	13:14
13	PL	Pelagic (tur)	13:14

Grain size ϕ class	
-6:-8	Cobble
-3:-5	Pebble
-2	Gravel
-1	Very Coarse Sand
0	Coarse Sand
1	Medium Sand
2	Fine Sand
3	Very Fine Sand
4	Silt
5	Silt
6	Silt
7	Silt
8	Clay
9:10	Clay
11:12	Clay
13:14	Clay

Figure 4.3: Top, the 13 facies simulated by Flumy grouped in $Nb = 2, 3, 7, 10, 13$. Bottom left Flumy facies with their corresponding grain size classes; and at the bottom right, the grain size classes description and colors.

The following parameters relate to the simulated domain.

- OX, OY: The location of the grid origin (also called *Longitude* and *Latitude* in Flumy-Userguide (2022));
- NX and NY: The number of grid nodes along the X and Y axis;
- DX and DY: The grid mesh size in meters along the X and Y axis;
- z_{min}: the bottom elevation in meters of the domain should correspond to the bottom surface of the reservoir (also called z_{ref} in Flumy-Userguide (2022));
- z_{max}: the top elevation in meters of the domain should correspond to the upper surface of the reservoir.
- DZ: the vertical interval of the domain $[z_{min}, z_{max}]$.

4.2.2.2 Simulated processes

At every iteration, the migration process occurs, while the aggradation and avulsion processes take place according to some fixed or random periods. All parameters described in this section can be modified at any time by pausing the simulation. Numerous parameters can be randomized, which is especially helpful to generate events whose occurrence is not exactly predictable. This stochastic approach permits to assess the *uncertainty* among all generated reservoirs. The whole parameters set description is available in Flumy-Userguide (2022).

Migration in Flumy. It is based on the evolution of the channel centerline in time and on the deposition of the associated sedimentary bodies. The $2D$ river hydraulic equations used

by the Flumy model proved to produce high-quality shapes (Lopez, 2003; Lopez et al., 2009). These equations come from the linearization of hydraulic equations proposed in the 1980s (Ikeda et al., 1981; Sun et al., 2001; Johannesson and Parker, 2013). These equations have assumptions, such as constant channel width, large curvature and steady state; however, these linear theories proved to give stable algorithms to model long-term evolution of meandering rivers. The mean flow velocity U depends on the global valley slope I and on the mean channel wavelength λ reached before first neck cutoff. The migration rate depends on the mean flow velocity U and the bank-erodibility coefficient E. The smaller this coefficient, the lower the erodibility of the bank, and the slower the migration of the meander. While the channel migrates, it incises the outer bank and deposits points bars in the inner bank. The channel centerline is discretized into several channel points. At their bank, a local velocity perturbation is calculated depending on U and E. When two meanders get closer due to their lateral migration, Flumy simulates a neck cutoff and instantly fills the oxbow lake with sand and mud plugs.

The main parameters associated with the migration are

- H_{max}: Channel bankfull maximum depth in meters;
- W: Channel bankfull width in meters;
- I: Global valley slope;
- λ: Mean wavelength reached before first neck cutoff in meters;
- E: Bank erodibility coefficient in meters per second;

According to field studies for fluvial meandering channelized systems, the channel aspect ratio W/H_{max} is usually equal to 10 (Lemay, 2018).

Aggradation in Flumy. The frequency and intensity of the overbank floods control the aggradation. Thickness and grain size of non-channelized material deposited decrease away from the channel, following a negative exponential distribution. Levee width corresponds to the grain size transition from silt to shale. The period and intensity of the overbank floods greatly influence the aggradation rate and the sand proportion of the resulted modeled reservoir. Additionally, Flumy has an option to perform vertical incisions in order to satisfy the *equilibrium profile* (Lopez et al., 2009) of the channelized system. In that case, the system turns into an erosive phase (no more aggradation). For the conditional simulations, this option will not be used, resulting in an aggrading system only.

The main parameters for aggradation are

- i_{ob}: Overbank flood intensity in meters. It can be constant or randomized following uniform/normal/log-normal distribution;
- T_{ob}: Overbank flood period [number of iterations]. It can be constant or randomized following a Poisson distribution;
- λ_{ob}: Overbank thickness exponential decrease in meters;
- L_{LV}: Levee width in meters.

Avulsion in Flumy. From time to time, a levee breach may occur, usually where the velocity perturbation is locally maximal in each meander. This results in the deposition of crevasse splays, possibly followed by an avulsion (new path downstream or chute cutoff). The abandoned path is instantly filled with sand and mud plugs. There are two types of avulsion events in the model:

- Local: a new path is tossed from a levee breach point within the simulated domain.

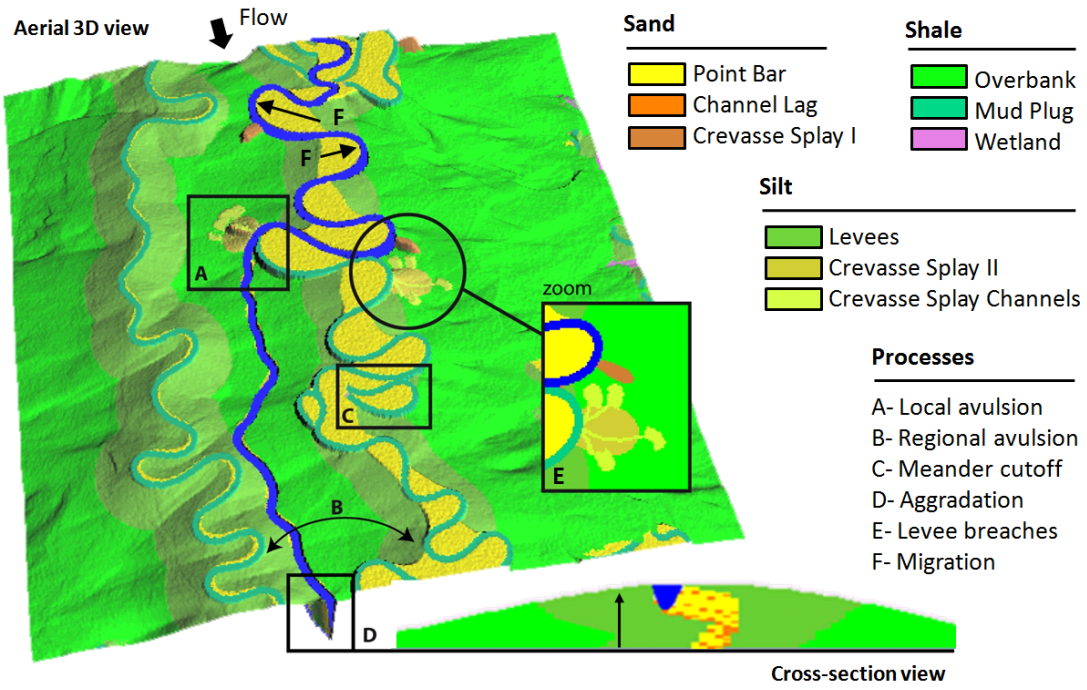


Figure 4.4: Illustration of main processes and facies (grouped by grain size class) in Flumy fluvial environment.

- Regional: a new path is tossed from a point which is upstream of the simulated domain. This ensures a uniform spatial distribution of the channel paths over the modeled domain through time.

The main parameters for avulsion processes are

- T_{LAV}: Local avulsion period [number of iterations]. It can be constant or randomized following a Poisson distribution;
- T_{RAV}: Regional avulsion period [number of iterations]. It can be constant or randomized following a Poisson distribution.

Fig. 4.4 shows a summary of the Flumy processes and main deposited facies.

4.2.3 Properties

4.2.3.1 Consolidated reservoir and unconsolidated zone

In Flumy, the reservoir erosion is controlled by the channel, either by the migration or by the avulsion processes (assuming that incisions option is not used, cf. Sect. 4.2.2.2, aggradation in Flumy). An *unconsolidated zone* exists at the top of the reservoir, where the channel may rework the previous deposits proportionally to the migration and aggradation rates ratio (Grimaud et al., 2022).

A Flumy simulation can be launched for a given number of iterations or by imposing an upper limit surface. For the last situation, the simulation stops when *the lowest elevation of the current topography reaches this upper limit*. Let z_{ul} be the elevation of the upper limit (the upper limit will always be considered flat in this work). If we are keen to work with a *consolidated* reservoir (which cannot be reworked afterwards up to the top of the reservoir at the elevation z_{max}), Eq. (4.1) must be satisfied:

$$z_{ul} = z_{max} + H_{max} \tag{4.1}$$

Fig. 4.5 illustrates the consolidated reservoir and unconsolidated zone concepts once z_{ul} elevation is reached by the whole topography.

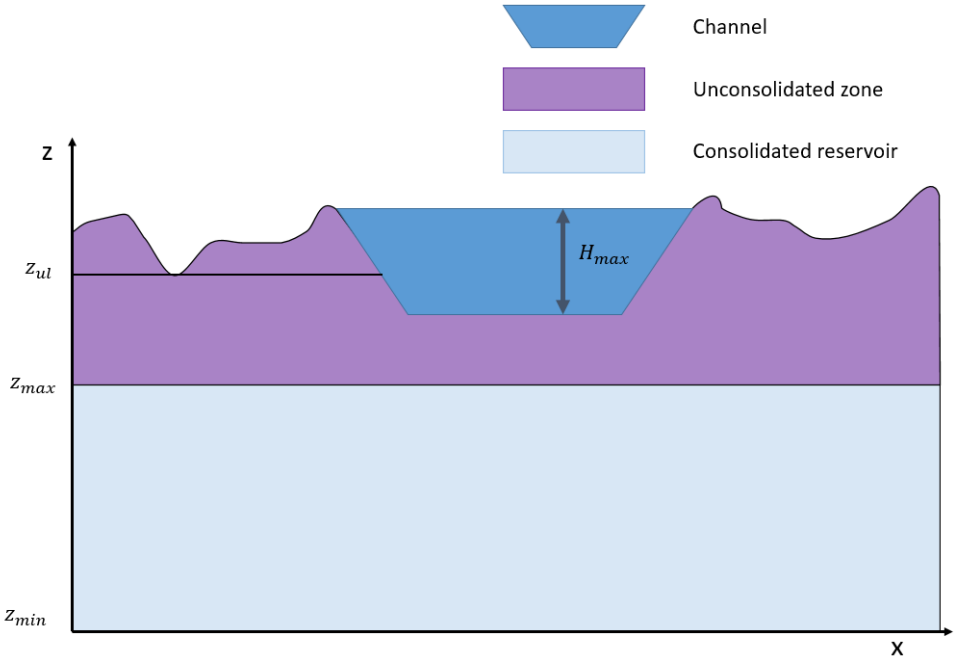


Figure 4.5: Vertical cross-section of a reservoir simulation. In light blue: the domain of interest within its vertical limits $[z_{min}, z_{max}]$. The simulation must be stopped at z_{ul} to ensure a consolidated reservoir. Above the vertical reservoir limit z_{max}, the unconsolidated zone, in purple, can be reworked by the migration and avulsion processes.

4.2.3.2 Migration and aggradation rates ratio

From Flumy parameters, forecasts for *migration* F_{mig} and *aggradation* F_{agg} rates are computed by the Flumy software (Flumy-Userguide, 2022). For the unconsolidated zone, the ratio between the migration and aggradation forecast rates allows computing the reworking rate r defined as:

$$r = \frac{F_{mig}}{F_{agg}} \tag{4.2}$$

When r increases, the migration predominates over the aggradation. As a consequence, the channel continually reworks the unconsolidated zone. On the contrary, the aggradation predominates over the migration. Consequently, the unconsolidated zone is more stable.

4.2.3.3 Sedimentary units and VPC

Before building the reservoir model, it is a common practice to estimate the facies proportion in the reservoir domain. This facies proportion is usually expressed in a *vertical proportion curve* (VPC), defining all the facies proportions in function of the vertical. A VPC can be often obtained in two ways: based on hard, or based on soft data. For the first, the VPC may be calculated as the mean proportion provided by the available wells in stacked intervals. Nevertheless, this solution carries some issues related to the representativity of a global variable from local ones. For the second, seismic images with geological interpretations can be used. Despite the global representation of these techniques, the uncertainty related to these interpretations may be important.

The VPC can be used to determine the different *sedimentary units* within the reservoir (Bubnova et al., 2020). The sedimentary units are bodies of rocks, bedded or unbedded, that are defined and characterized on the basis of their lithologic properties and their sedimentary relations. A reservoir is formed by either one or several stacked sedimentary units.

To reproduce them in Flumy, the reservoir is simulated, changing the parameters each time the topography reaches the bottom surface of the next sedimentary unit. Let $z_{i,max}$ be the top

surface of the sedimentary unit i. In order to reproduce the sedimentary unit i, the simulation is paused when the lowest topography reaches $z_{i,ul} = z_{i,max} + H_{i,max}$ ($H_{i,max}$: maximum channel depth for the sedimentary unit i). Hence, a simplification is considered: the limits of the sedimentary units are constant (horizontal surface).

If the reservoir is composed of more than one sedimentary unit, the inferior part of all but the first unit may contain some deposits simulated using parameters of the previous sedimentary unit. Indeed, since the unconsolidated zone can be reworked later, this zone is prone to be eroded. This decision privileges the consolidation of the upper part of the sedimentary units compared to their lower parts.

Remark If the considered facies is only the sand facies ($Nb = 2$), the VPC is called the sand VPC, which describes the sand proportion as a function of the vertical.

4.2.3.4 Impact of the processes on resulting global sand proportion

The relationship between the resulting sand proportion in the domain (also called *Net-to-Gross* N/G) and the physical processes in Flumy is the following.

- The migration process deposits points bars (sand facies) in the inner bank of the meanders, while it erodes its outer bank. As a consequence, the N/G increases when this process is more intense than the others. This process improves the horizontal connectivity of sand bodies in the reservoirs;
- The aggradation process produces fine-grained deposits resulting from overbank floods over the domain. As a consequence, the N/G decreases when this process is more intense than the others. Additionally, it improves the vertical connectivity of sand bodies in the reservoirs;
- The avulsion process produces abandoned channel facies. This process creates disconnected sand bodies and increases the volume of mud plugs and sand plugs related to abandoned channels.

4.2.4 Reference simulation

A synthetic case for fluvial environment is now presented. It is a non-conditional simulation, called the *reference simulation*, having only one sedimentary unit from which data, to be used later as conditional constraints, will be extracted. The following specific Flumy parameters values are used (all the other parameters are the default values shown in the Flumy software userguide (Flumy-Userguide, 2022)):

- OX and OY: 0×0m;
- NX and NY: 251×251;
- DX and DY: 10×10m;
- $DZ = [0, 10]$m;
- $H_{max} = 3$m;
- λ : 420m.

This means that the domain size is 2510×2510m and the thickness of the domain is 10m. The facies proportions are listed in Table 4.1.

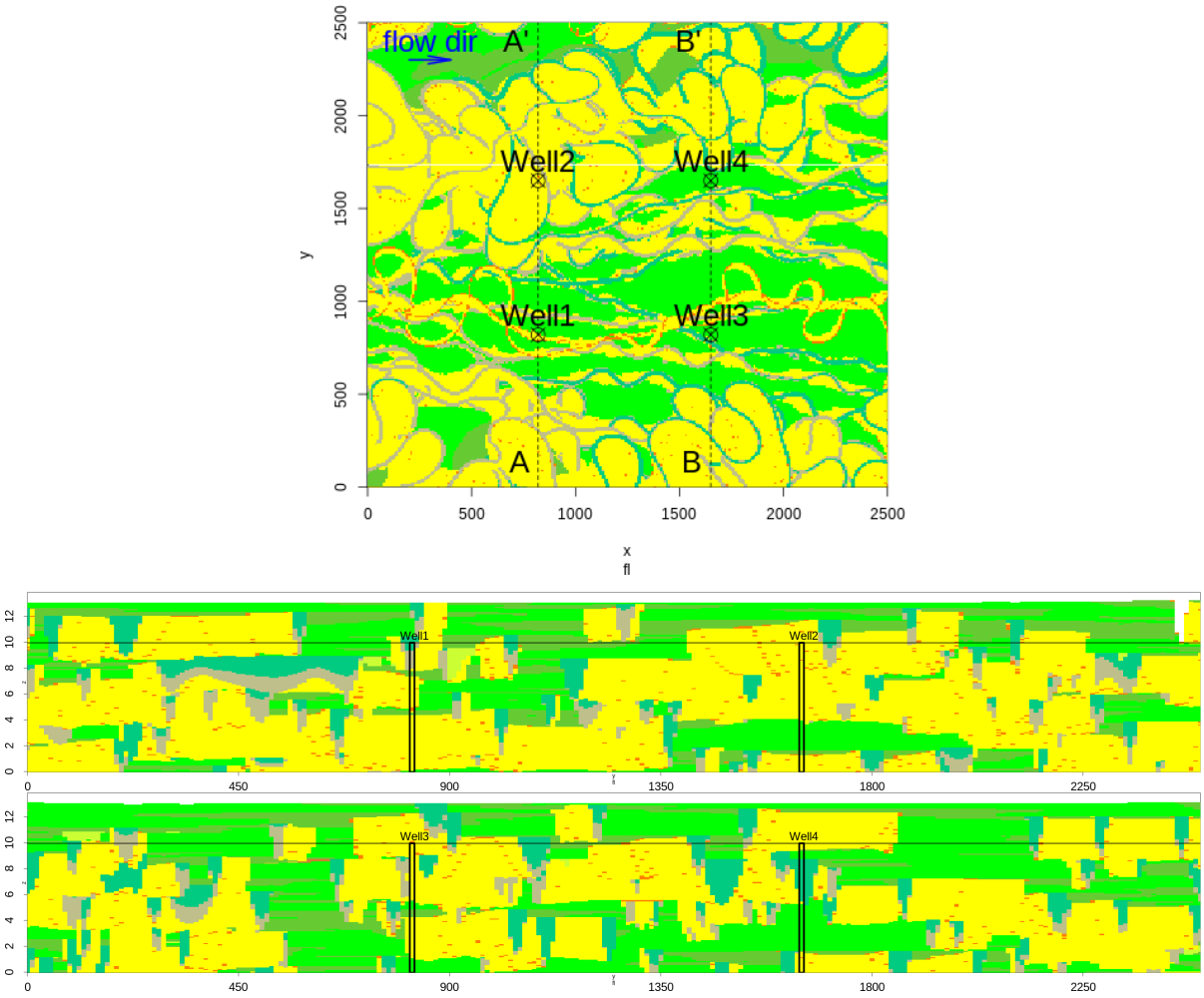


Figure 4.6: The **reference simulation** from where the pseudo-wells and sand VPC are extracted. Top, the horizontal slice ($z = 5m$), in the middle and on the bottom, the vertical sections $A - A'$ and $B - B'$, respectively. Color scales in Fig. 4.3.

Table 4.1: Percentage of Flumy facies for $Nb = 13$ and $Nb = 2$ for the reference simulation. Facies names in Fig. 4.3

Facies	CL	PB	SP	CS1	CCH	CS2	LV	OB	MP	WL	DR
%	1.5	52.5	7.0	0.0	0.2	0.0	9.1	22.2	7.5	0	0
%	61.0				39.0						

The N/G of the reservoir is 61.1%. Whereas the proportion of point bar is the most abundant for the sand facies (86.1% in proportion), the overbank is the most abundant for the clay facies (56.9% in proportion).

Fig. 4.6 shows a horizontal section in the middle of the domain ($z = 5m$), and two vertical sections $A - A'$ and $B - B'$ of the reference simulation for the thirteen-facies wells ($Nb = 13$) ($x = 830$m and $x = 1660$m). Fig. 4.7 shows a $3D$ view.

4.3 Conditional simulation

4.3.1 Definition of the constraints to be honored

Now, let us try to condition the Flumy model to field data constraints. Field data usually comes in two forms, *hard data* and *soft data*. We are interested in the reservoir simulation considering

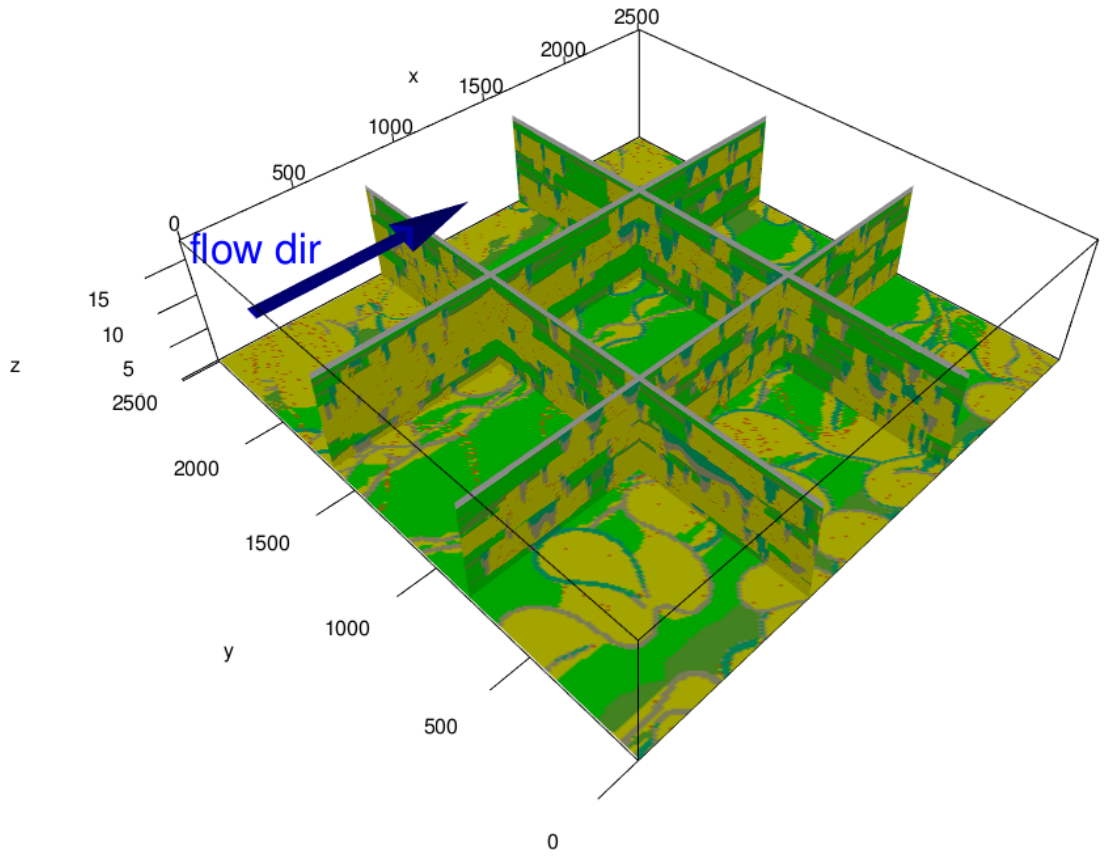


Figure 4.7: $3D$ view of the **reference simulation**.

Table 4.2: Wells location

Well	**X m**	**Y m**
Well 1	830	830
Well 2	830	1660
Well 3	1660	830
Well 4	1660	1660

these two different types of constraints. We work with facies information, along the wells and coming from VPC.

From the reference simulation, two types of information are extracted:

- Four pseudo-wells with $Nb = 2$ and $Nb = 13$ facies;
- The sand VPC computed from the whole $3D$ block of facies as if it were resulting from seismic interpretation.

These information will be used as *facies observations* for testing the conditioning methods. Additionally, the grain size classes are extracted for the same wells. The location of the 4 extracted wells is presented in Table 4.2. The reference conditioning data for the synthetic case are shown in Fig. 4.8.

4.3.2 Criteria to assess the match between observations and simulated model

To measure the quality of the conditioning methods, a series of criteria are created. Due to the number of notations used from here, they have been listed in Appendix B.3.

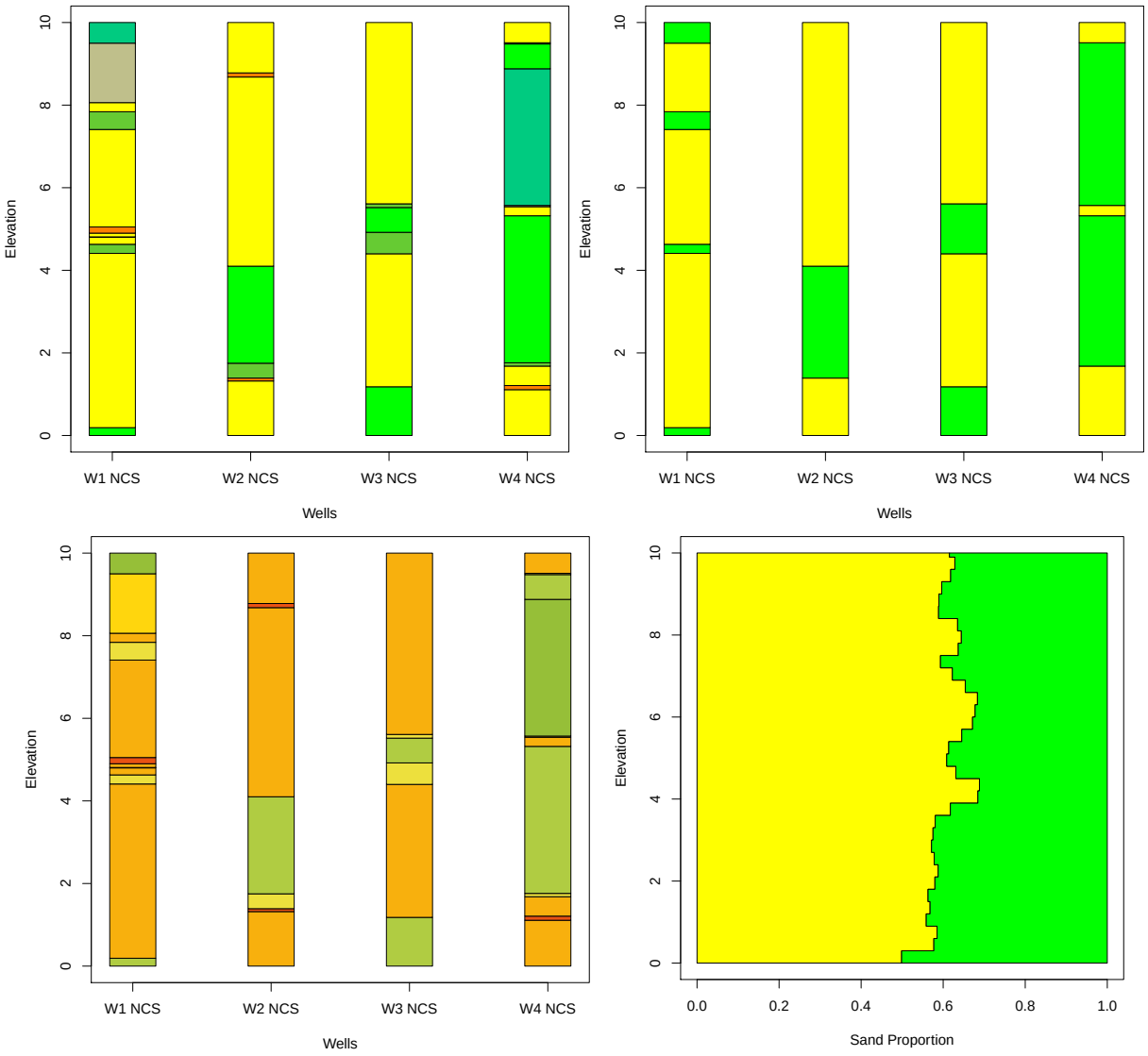


Figure 4.8: Conditioning data extracted from the reference simulation: wells with all 13 facies (top left), wells with facies grouped in 2 categories (top right), their grain size (bottom left) and the sand VPC (bottom right) computed from the whole $3D$ block. Color scales in Fig. 4.3.

4.3.2.1 Well facies match

Let us assume that L wells are available. Let $F_\ell(z)$ be the facies observed in well ℓ at elevation z. We would like to compare it to its simulated counterpart $F_\ell^{(j)}(z)$, deposited by a simulation denoted as j.

The *facies match* between data and a simulation j of well ℓ in a generic interval $I = [zb, zt]$ is:

$$FM_{I,\ell}^{(j)} = \frac{1}{zt - zb} \int_{zb}^{zt} \mathbb{1}_{F_\ell(z)=F_\ell^{(j)}(z)} \, dz \tag{4.3}$$

In the case the data or the simulation would not be informed from zb to zt, the expression should be modified by the more general form:

$$FM_{I,\ell}^{(j)} = \frac{\int_{zb}^{zt} \psi_\ell^{(j)}(z) \mathbb{1}_{F_\ell(z)=F_\ell^{(j)}(z)} \, dz}{\int_{zb}^{zt} \psi_\ell^{(j)}(z) \, dz} \tag{4.4}$$

with $\psi_\ell^{(j)}(z)$ a weight function depending on z. Let $\psi_\ell^{(j)}(z)$ be expressed as:

$$\psi_\ell^{(j)}(z) = \mathbb{1}_{max\{z_{b,\ell}, z_{b,\ell}^{(j)}, zb\} \leq z \leq min\{z_{t,\ell}, z_{t,\ell}^{(j)}, zt\}} \tag{4.5}$$

with $z_{b,\ell}, z_{t,\ell}$ the elevation limits of well ℓ, and $z_{b,\ell}^{(j)}, z_{t,\ell}^{(j)}$ the elevation limits of the pillar extracted at well ℓ location from a simulation j.

Considering the domain vertical interval $DZ = [z_{min}, z_{max}]$, the *facies match* between data and a simulation j of well ℓ in the domain is:

$$FM_{DZ,\ell}^{(j)} = \frac{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z) \mathbb{1}_{F_\ell(z)=F_\ell^{(j)}(z)} \, dz}{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z) \, dz} \tag{4.6}$$

with $\psi_\ell^{(j)}(z)$ defined by Eq. (4.5).

A measure of the facies match for a simulation j can be obtained, considering simultaneously the L wells. Two alternatives are then proposed, the *weighted average facies match* and the *minimum facies match* over all wells in a vertical interval I.

Let $FM_{I,average}^{(j)}$ be the average facies match of the L wells depending on the well lengths for an interval I on a simulation j.

$$FM_{I,average}^{(j)} = \sum_{\ell=1}^{L} FM_{I,\ell}^{(j)} \frac{\int_{zb}^{zt} \psi_\ell^{(j)}(z) \, dz}{\sum_{\ell'=1}^{L} \int_{zb}^{zt} \psi_{\ell'}^{(j)}(z) \, dz} \tag{4.7}$$

with $\psi_\ell^{(j)}(z)$ defined by Eq. (4.5). On the other hand, let $FM_{I,min}^{(j)}$ be the minimum facies match over the L wells on a simulation j in a vertical interval I.

$$FM_{I,min}^{(j)} = \min_{1 \leq \ell \leq L} FM_{I,\ell}^{(j)} \tag{4.8}$$

If the domain vertical interval $DZ = [z_{min}, z_{max}]$ is considered, the weighted average facies match over the L wells on a simulation j in the domain is the following:

$$FM_{DZ,average}^{(j)} = \sum_{\ell=1}^{L} FM_{DZ,\ell}^{(j)} \frac{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z) \, dz}{\sum_{\ell'=1}^{L} \int_{z_{min}}^{z_{max}} \psi_{\ell'}^{(j)}(z) \, dz} \tag{4.9}$$

4.3.2.2 Well grain size class match

Let us assume that L wells are available. The grain size class data is an integer variable ranging from 1 to N_{Class}, the number of classes used to discretize the distribution of the grain size. A value of 1 correspond to coarse grains (i.e., cobble), and a value of N_{Class} corresponds to the finest grains (i.e., clay). Let $G_\ell(z)$ be the grain size class data at elevation z for well ℓ. The grain size discretization in Flumy is 16, hence $N_{Class} = 16$. Therefore, we would like to compare it to its simulated analogue, defined as $G_\ell^{(j)}(z)$ of a simulation j.

The *well grain size class match* between data and simulation j for well ℓ in the reservoir domain is defined by:

$$GM_\ell^{(j)} = \frac{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z)[1 - \frac{|G_\ell(z) - G_\ell^{(j)}(z)|}{N_{Class} - 1}]\, dz}{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z)\, dz} \tag{4.10}$$

with $\psi_\ell^{(j)}(z)$ defined by Eq. (4.5).

A grain size class match between data and a simulation j can be then obtained, considering simultaneously the L wells. Let $GM_{average}^{(j)}$ be the average grain size class match of the L wells depending on the well lengths in the reservoir domain on a simulation j.

$$GM_{average}^{(j)} = \sum_{\ell=1}^{L} GM_\ell^{(j)} \frac{\int_{z_{min}}^{z_{max}} \psi_\ell^{(j)}(z)\, dz}{\sum_{\ell'=1}^{L} \int_{z_{min}}^{z_{max}} \psi_{\ell'}^{(j)}(z)\, dz} \tag{4.11}$$

with $\psi_\ell^{(j)}(z)$ defined by Eq. (4.5).

4.3.2.3 Sand proportion match

In this section, facies proportions are presented in different supports. For all the formulae, it is assumed that a facies is chosen to obtain its proportion. Let $FP(z)$ be the *facies proportion* of data at elevation z. For a simulation j, let $FP^{(j)}(z)$ be the *facies proportion* at elevation z.

The facies proportion can be calculated in intervals. Let FP_I be a facies proportion of data in a generic interval $I = [zb, zt]$. Let $F_I^{(j)}$ be a facies proportion for a simulation j in the same interval.

$$FP_I = \int_{zb}^{zt} FP(z)\, dz \tag{4.12}$$

$$FP_I^{(j)} = \int_{zb}^{zt} FP^{(j)}(z)\, dz \tag{4.13}$$

The *facies proportion match* between data and a simulation j in the vertical interval I is defined as:

$$FPM_I^{(j)} = 1 - \left| FP_I^{(j)} - FP_I \right| \tag{4.14}$$

Similarly, let FP_{DZ} be the facies proportion of data in the domain, and let $FP_{DZ}^{(j)}$ be the facies proportion for a simulation j in the same interval.

$$FP_{DZ} = \int_{z_{min}}^{z_{max}} FP(z)\, dz\,, \tag{4.15}$$

$$FP_{DZ}^{(j)} = \int_{z_{min}}^{z_{max}} FP^{(j)}(z)\, dz \tag{4.16}$$

The facies proportion match between data and a simulation j in the domain is defined as:

$$FPM_{DZ}^{(j)} = 1 - \left| FP_{DZ}^{(j)} - FP_{DZ} \right| \tag{4.17}$$

On the other hand, the *facies proportion balance* between data and a simulation j in the domain is defined as:

$$FPB^{(j)} = 1 - \left(FP_{DZ}^{(j)} - FP_{DZ}\right) \tag{4.18}$$

Consider the case of the sand facies. If $FPB^{(j)} < 1$, the simulation contains more sand than the data, and conversely if $FPB^{(j)} > 1$. The average over a number of simulations of Eq. (4.18) says if the simulations have (on average) more or less sand than data. By contrast, the average of Eq. (4.17) says how well the proportion of sand in data is honored by the simulations.

In this thesis, for the analysis of synthetic and real cases, the *facies proportion* considered is only sand ($Nb = 2$). Therefore, the (global or by-layer) facies proportion (and by extension, the facies VPC) will be also called *sand proportion* (and by extension, sand VPC) in this manuscript.

Remark The facies proportion match and facies proportion balance in an interval are less selective than the well facies and grain size class matches, which measure matches punctually along the interval.

4.3.2.4 Selection criteria among many simulations

When a set of conditional simulations $j = 1, \ldots, p$ for a reservoir is available, different criteria can be applied to select a subset which better honors the data. Moreover, when a set of independent conditional simulations is available, the uncertainty can be assessed. Let us assume that a number of conditional simulations is available. Using formulae (4.9) and (4.18), as a *quality measure*, a simulation subset can be obtained. Let FM, FPM, and R three simulations chosen from the simulation set:

- FM is the simulation which has the highest weighted average facies match in the domain.

$$FM = \underset{1 \leq j \leq p}{\arg\sup}\, FM_{DZ}^{(j)} \tag{4.19}$$

- FPM is the simulation which has the sand proportion balance closest to the unity.

$$FPM = \underset{1 \leq j \leq p}{\arg\min}\{|FPB^{(j)} - 1|\} \tag{4.20}$$

- R is a simulation chosen randomly:

$$R \sim \mathcal{U}\{1, \ldots, p\} \tag{4.21}$$

All the ingredients are presented. Now, it is time to perform conditional simulations and compare the dynamic algorithm to the sequential one proposed in the thesis.

4.3.3 Dynamic conditioning method

4.3.3.1 Background

This method is the one currently implemented in Flumy (Bubnova, 2018). The conditioning is based on each well data that is active depending on the elevation and the position of the meandering channel at the current iteration. The channel is attracted to or repelled from the wells depending on the current active facies (the one situated at the well location to the same elevation as the current topography). Each time the topography is modified at a well location (aggradation or incision), Flumy validates the deposits that either match the data or cannot be later replaced by a matching facies. In this thesis, only well data constraints will be considered for testing the dynamic conditioning algorithm. Indeed, the dynamic conditioning method is not able to honor sand VPC.

The Flumy fluvial facies for the dynamic conditioning method are classified in three groups according to their deposited environment:

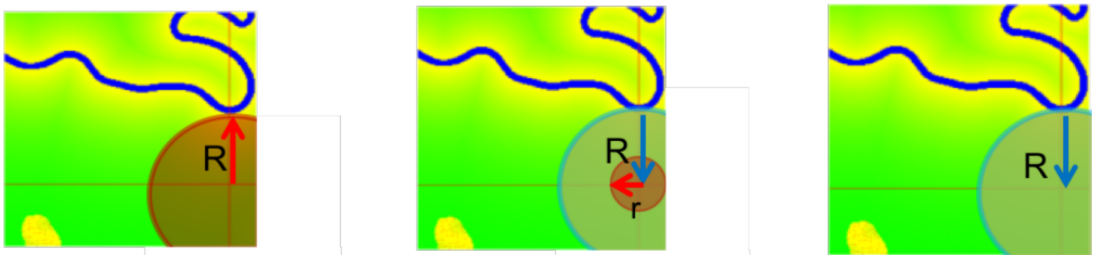


Figure 4.9: Main principles of migration adaptation in Flumy dynamic conditioning method (here R and r are internal radius parameters)(figure taken from Bubnova (2018))

- Channelized facies: channel lag (CL), point bar (PB), sand plug (SP), mud plug (MP);
- Levee facies: crevasse splay I (CSI), crevasse splay II (CSII), crevasse channel (CCh) and levee (LV);
- Non-Channelized facies: overbank alluvium (OB), draping (DR) and wetland deposit (WL).

A second classification aims to identify the validated deposits that must not be eroded until the end of the simulation.

- "Erodable" facies: Channelized facies: channel lags (CL), point bar (PB), sand plug (SP), mud plugs (MP). These facies are all channelized facies. These facies are deposited by the migration/avulsion processes. They can be eroded since they will necessarily be replaced by equivalent facies. If an "erodable" facies happens to be deposited at a well, it will be validated even if it does not honor the data because honoring it afterwards will not be possible.
- "Not-to-be-eroded" facies: overbank alluvium (OB), wetland deposits (WL), crevasse splay facies (CSI, CSII and CCh), draping (DR) and levees (LV). These facies are deposited by the aggradation process and during crevasse splays maturation or marine incursion. Such facies in well data will be honored by the deposition of a facies of the same category and must not be eroded later.

Here is a quick description of the dynamic conditioning algorithm. More details of this method can be found in Bubnova (2018).

Adaptation of the Migration. At a given time, each well is linked to the closest meander. According to the active conditioning facies (the one located at the same elevation than the current topography at well), the velocity perturbation of the points of the connected meander is adapted as follows:

- Active clay facies: the meander migration is reduced to repel the channel from the well. Thus, no channelized facies will be deposited at well.
- Active levee facies: the meander migration is increased to attract the channel to the well or reduced if it is too close to the well. Thus, levees facies could be deposited at well.
- Active channelized facies: the meander migration is increased to attract the channel at well. Then, the previous deposited facies could be replaced by sand at well. When sand is urgently required at well, the aggradation could be temporarily blocked in order to let the meander reach the well.

Fig. 4.9 shows the adaptation of the migration.

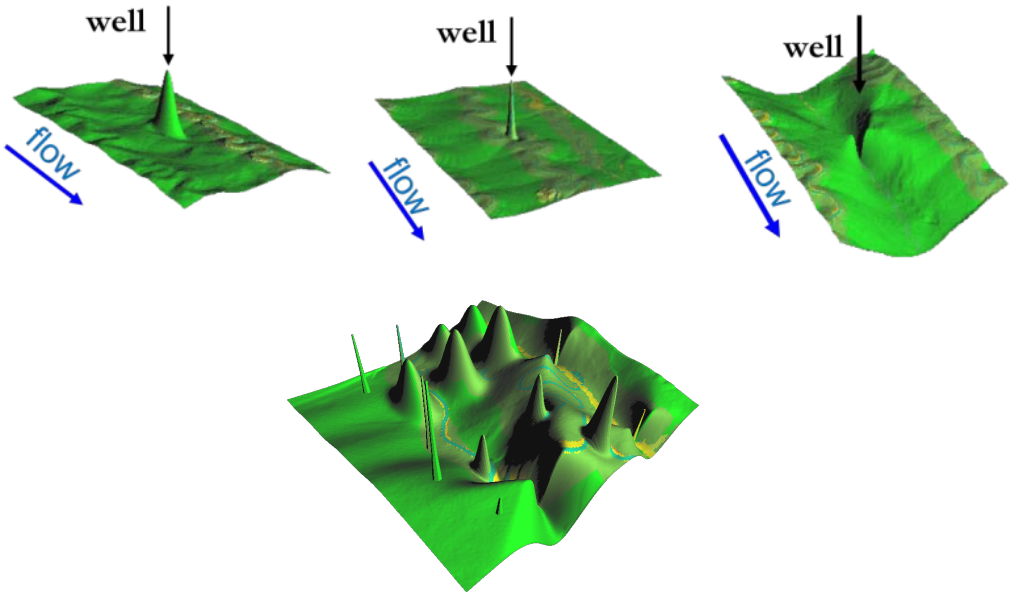


Figure 4.10: Top, modified virtual topographies used to adapt the avulsion process in Flumy. From left to right, a well with an active clay facies, with an active levee facies, and with an active channelized facies. Bottom, a modified virtual topography with many wells combining different deformations.

Adaptation of the Avulsions. When an avulsion occurs, it is mandatory that the channel does not erode the validated non-channelized facies at the well. To avoid this phenomenon, the new channel path is tossed on a virtual deformed topography according to the active facies in each well. The applied deformations can be seen in Fig. 4.10. If the well contains an active clay facies, a big bump is created at well location and the new channel will avoid it. When a levee facies is active, a rimmed bump is created at well location in order to attract the new channel close to but not on the well. Finally, a valley is formed to attract the channel at well position when the well contains an active channelized facies.

4.3.3.2 Results in the case of two-facies

Considering the wells with two facies sand-clay ($Nb = 2$) from the reference simulation (Fig. 4.8) as conditional data, a number of 250 independent conditional simulations are performed. The thirteen deposited facies are then grouped into the same two-facies categories as the conditioning data. Fig. 4.11 shows the results in box plots of the two-facies match for each well (Eq.(4.6)), their average facies match by the Eq. (4.9), their sand proportion balance (Eq. (4.18)), and their sand proportion match (Eq. (4.17)). Additionally, it shows the sand VPC of the reference simulation (in red) and the sand proportion of each conditional simulation (Eq. (4.26)) (in gray) with their average (in black). The average value of the well facies match of the 250 independent realizations is 85.15%. The sand proportion balance in the domain is 101.97%. It shows that, on average, simulations have 1.97% less sand than in data. The sand proportion match in the domain shows, on average, 97.24%of the sand observed at wells.

Using the criteria from expressions (4.19), (4.20), and (4.21), three simulations are selected among the 250 independent realizations. Their match results by well are shown top left in Fig. 4.12. The simulation FM is chosen, and different sections are shown in the same figure. Fig. 4.13 shows a $3D$ view of the same simulation. In the middle of the *Well 4*, a small sand thickness should have been deposited.

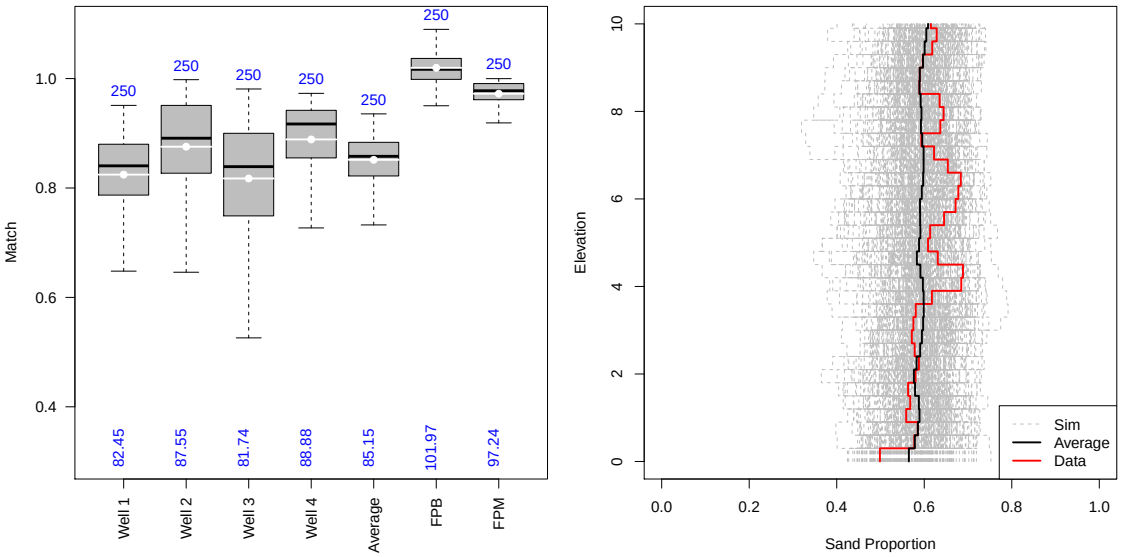


Figure 4.11: Results of the **dynamic conditioning method** with the data and conditional simulations grouped in **two-facies wells** for the **synthetic case** from 250 independent realizations. Left, box plots of the facies and grain size class matches for the 4 wells, their average, the sand proportion balance and the sand proportion match. Blue values on the top are the number of simulations, the blue values on the bottom recall the match value in %. Right, the sand VPC data (sand VPC of the reference simulation) in red, the sand VPC of simulations in gray and their average in black.

4.3.4 Sequential conditioning method

This section presents the novel method which performs a conditional simulation of Flumy model using particle filtering. The method rests on a sequential assimilation of the conditioning data, which can be not only the well facies, but also the sand VPC.

4.3.4.1 Stacked layers for sequential adaptation

From a geometrical perspective, Flumy fills the block model from base to top of the reservoir. Considering the aggrading system, this progressive building procedure can be seen as a succession of k sediments layers stacked one onto the other. By using the capability of Flumy to be paused when the topography exceeds a specific upper limit surface, a Flumy simulation can be split into several steps, where each step fills a consolidated layer of sediments.

We define a layer $i \in \{1..., k\}$ by its bottom elevation z_i and its top elevation z_{i+1}. For a complete reservoir, we have $z_1 = z_{min}$ and $z_{k+1} = z_{max}$. Remember that the upper part of the current simulation can be later reworked due to the avulsions and the lateral migration of the channel (cf. sect 4.2.3). Hence, in order to ensure that the layer i is consolidated, its upper limit $z_{i,ul}$ must be equal to $z_{i+1} + H_{max}$ (cf. Sect. 4.2.3.1). Fig. 4.14 illustrates these different elevations for a given step i.

Executing one step i depends solely on the model parameters and the initial state of step i (which corresponds to the final state of the previous step $i-1$). Thus, this is a sequential process that permits to assimilate sequentially the data constraints using particle filtering (cf. Sect. 2.3).

The thickness of the layers must be chosen. On one hand, if the layers are very thin, the precision of the algorithm is better since the match criteria are calculated for smaller intervals but the number of steps increases, scaling up the computing time. On the other hand, if the layers are very thick, the facies match criteria are less selective, but the computing time is lower with fewer steps. In this work, we have chosen to make this thickness related to the channel maximum depth at $H_{max}/10$, the final layer being smaller.

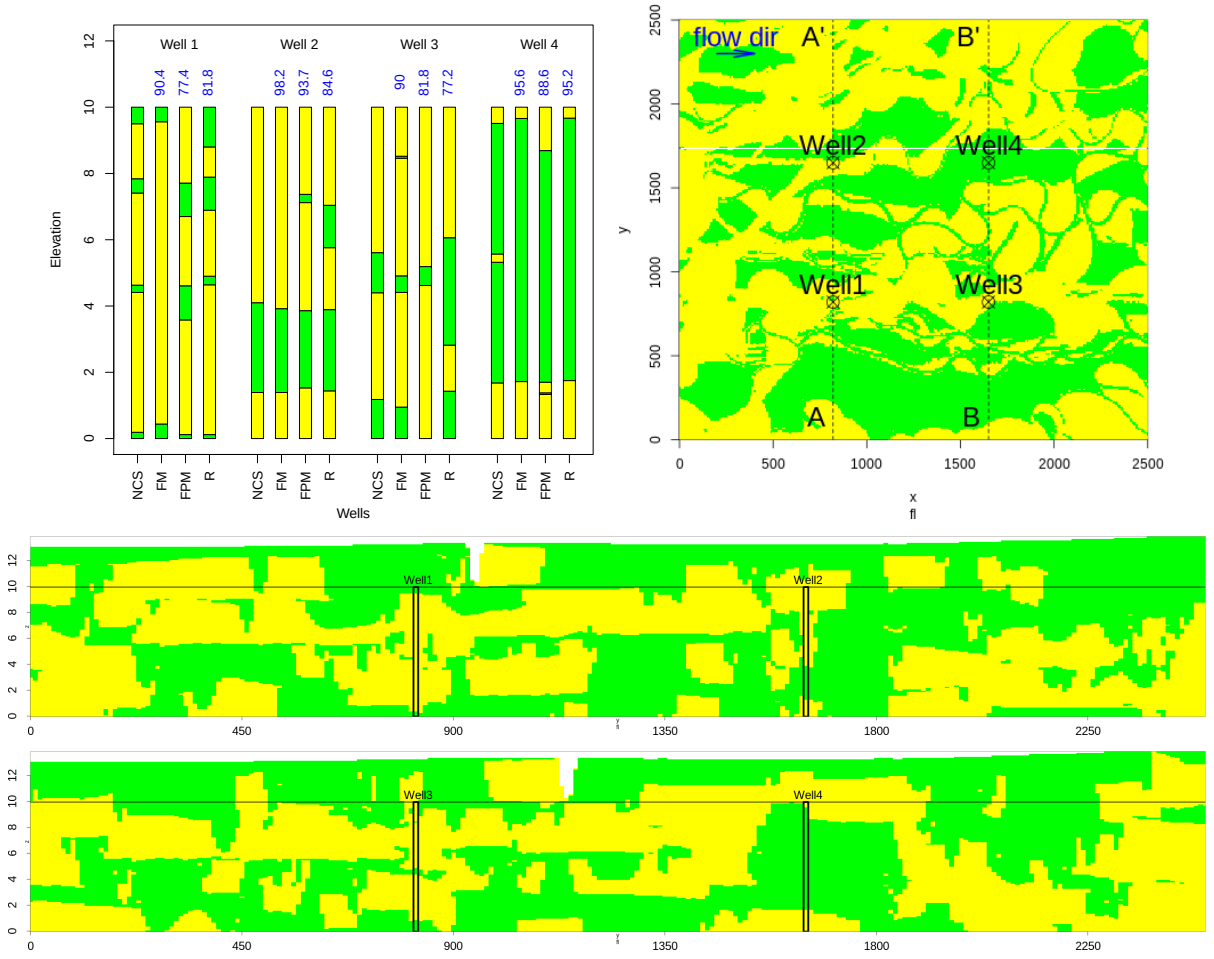


Figure 4.12: From 250 simulations of the **dynamic conditioning method** with **two-facies wells** for the **synthetic case**, three simulations are selected using the three criteria from Eqs. (4.19), (4.20) and (4.21) shown on the top left. Top right, the horizontal slice (z=5m) of the Flumy simulation FM. In the middle and on the bottom, the vertical sections $A - A'$, and $B - B'$, respectively.

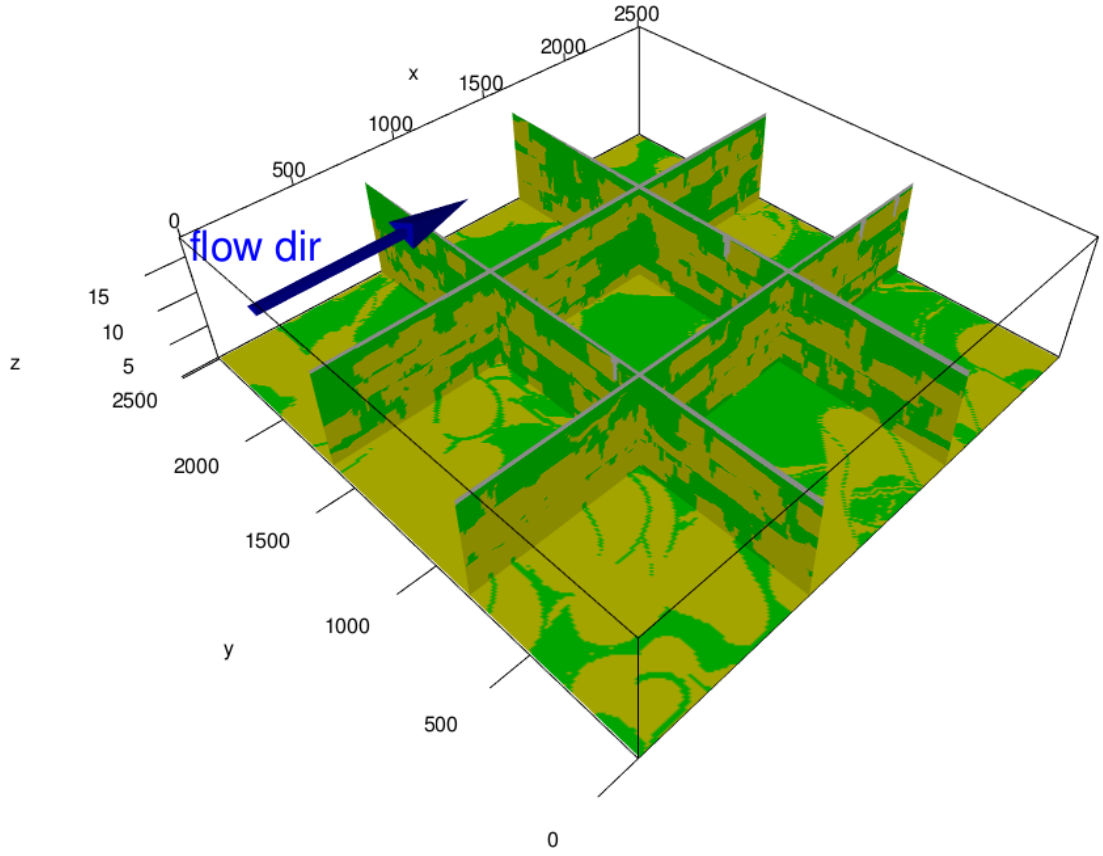


Figure 4.13: $3D$ view of a realization for the FM criterion using the **dynamic conditioning method** with **two-facies wells** for the **synthetic case**.

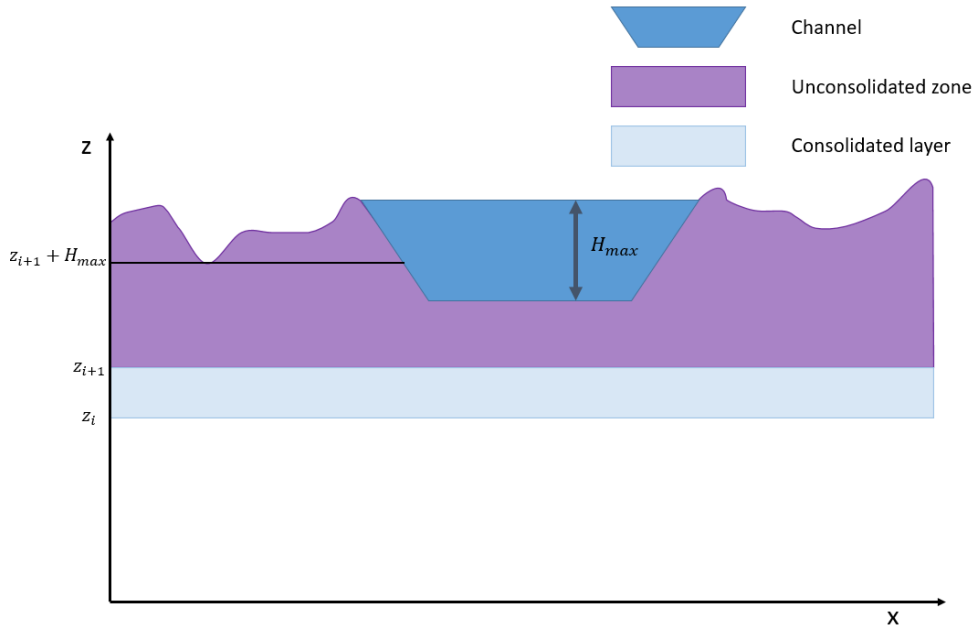


Figure 4.14: Scheme of a sequential Flumy simulation of the current step i (vertical section). In light blue, the last consolidated layer, in purple, the unconsolidated zone.

4.3.4.2 Well facies match in consolidated layer

Similarly to Sect. 4.3.2.1, we can define the facies match between data and a simulation j of well ℓ in the step i in its consolidated layer using Eq. (4.4). For the sake of simplification, we denote as i the interval $[z_i, z_{i+1}]$ for the consolidated layer.

$$FM_{i,\ell}^{(j)} = \frac{\int_{z_i}^{z_{i+1}} \psi_\ell^{(j)}(z) \mathbb{1}_{F_\ell(z)=F_\ell^{(j)}(z)}\, dz}{\int_{z_i}^{z_{i+1}} \psi_\ell^{(j)}(z)\, dz} \tag{4.22}$$

with $\psi_\ell^{(j)}(z)$ defined by Eq. (4.5).

A measure of the facies match for a simulation j can be obtained, considering simultaneously the L wells for the consolidated layer. Two alternatives are then proposed, the *weighted average facies match* and the *minimum facies match* over all wells at step i.

The average facies match of the L wells depending on the well lengths at step i on a simulation j is:

$$FM_{i,average}^{(j)} = \sum_{\ell=1}^{L} FM_{i,\ell}^{(j)} \frac{\int_{z_i}^{z_{i+1}} \psi_\ell^{(j)}(z)\, dz}{\sum_{\ell'=1}^{L} \int_{z_i}^{z_{i+1}} \psi_{\ell'}^{(j)}(z)\, dz} \tag{4.23}$$

On the other hand, the minimum facies match over the L wells on a simulation j at step i in its consolidated layer is:

$$FM_{i,min}^{(j)} = \min_{1\leq\ell\leq L} FM_{i,\ell}^{(j)} \tag{4.24}$$

The facies proportion of data and for a simulation j at step i can be defined as:

$$FP_i = \int_{z_i}^{z_{i+1}} FP(z)\, dz\,, \tag{4.25}$$

$$FP_i^{(j)} = \int_{z_i}^{z_{i+1}} FP^{(j)}(z)\, dz \tag{4.26}$$

Then, facies proportion match between data and a simulation j at step i is defined as:

$$FPM_i^{(j)} = 1 - \left| FP_i^{(j)} - FP_i \right| \tag{4.27}$$

At the end of each step, the corresponding consolidated layer is totally computed, under the assumption that no vertical incisions can occur during the reservoir evolution. The unconsolidated zone is available and stored to be used as initial state for the next step. However, this unconsolidated zone could also provide useful forecast information for the next step deposits.

4.3.4.3 Well facies match in unconsolidated zone

Since the creation of an unconsolidated zone is mandatory to obtain the consolidated layer i, we take advantage of its existence as a *prediction*. However, this zone is prone to be eroded, and this forecast is not necessarily true. Therefore, the probability that validated data at a given well may be eroded depends on the ratio between the migration and aggradation rates r (cf. sect. 4.2) and on the distance between this validated data and the bottom surface of the unconsolidated zone.

The unconsolidated zone is used to measure the well facies matches by an expression based on Eq. (4.22). For the sake of simplification, we denote as i the interval $[z_{i+1}, z_{i+1} + H_{max}]$ for the unconsolidated zone.

$$FMU_{i,\ell}^{(j)} = \frac{\int_{z_{i+1}}^{z_{i+1}+H_{max}} \psi_\ell^{(j)}(z) \mathbb{1}_{F_\ell(z)=F_\ell^{(j)}(z)}\, dz}{\int_{z_{i+1}}^{z_{i+1}+H_{max}} \psi_\ell^{(j)}(z)\, dz} \tag{4.28}$$

with $\psi_i^{(j)}(z)$ a function of z and j a given simulation:

$$\psi_{i,\ell}^{(j)}(z) = exp\left(-r\mu[z - z_{i+1}]\right) \mathbb{1}_{max\{z_{b,\ell}, z_{b,\ell}^{(j)}, z_{i+1}\} < z < min\{z_{t,\ell}, z_{t,\ell}^{(j)}, z_{i+1} + H_{max}\}}$$

including a coefficient μ in the decreasing exponential function of r (cf. sect. 4.2.3.2).

A measure of the facies match for a simulation j can be obtained, considering simultaneously the L wells for the unconsolidated zone. Two alternatives are also proposed, the *weighted average facies match* and the *minimum facies match* over all wells at step i.

The average facies match of the L wells depending on the well lengths at step i in a simulation j is:

$$FMU_{i,average}^{(j)} = \sum_{\ell=1}^{L} FMU_{i,\ell}^{(j)} \frac{\int_{z_i}^{z_{i+}} \psi_\ell^{(j)}(z)\, dz}{\sum_{\ell'=1}^{L} \int_{z_i}^{z_{i+}} \psi_{\ell'}^{(j)}(z)\, dz} \tag{4.29}$$

On the other hand, the minimum facies match over the L wells on a simulation j at step i in its unconsolidated zone is:

$$FMU_{i,min}^{(j)} = \min_{1 \le \ell \le L} FMU_{i,\ell}^{(j)} \tag{4.30}$$

4.3.4.4 Particle weight equation

For the particle filtering, it is mandatory to calculate the particle weights at the end of each step i. Then, particle weights are used for the resampling step (cf. Sect. 2.3). For a given consolidated layer at step i of a particle $j, (1 \le j \le p)$, we already know how to calculate the well facies matches $FM_{i,\ell}^{(j)}$ and $FMU_{i,\ell}^{(j)}$ in the consolidated layer and unconsolidated zone, respectively. Additionally, the facies proportion match at the step i $FPM_i^{(j)}$ is also available for the consolidated layer. A linear equation is then proposed for the *particle weight calculation*[1]:

$$w_i^{(j)} = \frac{\alpha FM_i^{(j)} + \beta FMU_i^{(j)} + \gamma FPM_i^{(j)}}{\sum_{j'=1}^{p} \left(\alpha FM_i^{(j')} + \beta FMU_i^{(j')} + \gamma FPM_i^{(j')}\right)} \tag{4.31}$$

with $\alpha, \beta, \gamma \ge 0$, $\alpha + \beta + \gamma = 1$ and where $FM_i^{(j)}$ and $FMU_i^{(j)}$ can be chosen from Eqs. (4.23) or (4.24) and (4.29) or (4.30) for the consolidated layer and the unconsolidated zone. In the case of a *punctual match* ($\alpha > 0, \beta > 0$ and $\gamma = 0$):

- Average criterion: Eqs. (4.23) and (4.29) are used. Then

  $$FM_i^{(j)} = FM_{i,average}^{(j)};\ FMU_i^{(j)} = FMU_{i,average}^{(j)}$$

  While the number L of wells increases, the criterion may lead to resample particles where the match is poor for some wells.

- Min criterion: Eqs. (4.24) and (4.30). Then

  $$FM_i^{(j)} = FM_{i,min}^{(j)};\ FMU_i^{(j)} = FMU_{i,min}^{(j)}$$

  While the number L of wells increases, there are more chances of having a well facies match null at step i for particle j. However, it makes the algorithm more selective, since it focuses on the worst facies match observed in wells.

[1] In the case that the three terms of the numerator are 0 for all the particles, the weights are conventionally set to 0, and the simulation stops.

- Perfect match criterion: An indicator function is used for each well ℓ when the facies match is 1 for the consolidated zone. Then, the product is computed along the L wells.

$$FM_i^{(j)} = \prod_{\ell=1}^{L} \mathbb{1}_{FM_{i,\ell}^{(j)}=1} \qquad (4.32)$$
$$\beta = 0$$

  This criterion ensures a 100% well facies match for all L wells at the end of the conditional simulation. The unconsolidated zone is negligible, since it can be reworked later. Nevertheless, this criterion is too strict and cannot be used in real applications. Indeed, the probability to deposit the matching facies at one well location and with the same thickness is extremely low.

When the weights are defined for the current step i, the resampling step is performed using the *residual resampling*, since it produces results with smaller variance (cf. sect 2.3.2).

4.3.4.5 Particle selection

At the end of the conditional simulation using the sequential conditioning method, p candidate simulations are available. According to the algorithm 4 from Sect. 2.3, a candidate must be chosen randomly to represent the conditional simulation. However, from an operational perspective, not only one candidate can be considered, but also a set of candidates according to certain criteria. For example, three candidates: FM, FPM, and R can be selected from Eqs. (4.19), (4.20) and (4.21), respectively.

4.3.4.6 Sensitivity analysis

A sensitivity analysis has been performed to measure the impact of the different parameters: μ (cf. Eq. (4.28)), p (number of particles), weight criteria (Average or Min), α, β, and γ (cf. Eq. (4.31)) for the *two facies* wells. A number of 25 independent realizations have been performed using the sequential conditioning method: see results in Sect. 4.3.4.7. Several parameter combinations have been tested. More details are provided in Appendix B.1. The conclusions are the following:

- Using the sand facies proportion match per layer (Eq. (4.27)) does not improve the global reproduction of the sand facies proportion. Consequently, γ can be neglected;
- The Min criterion per layer (Eq. (4.24)) gives better results than the Average criterion per layer (Eq. (4.23)) regarding the well facies match at the end of the realizations;
- Increasing the number of particles improves the mean well match (Eq. (4.9));
- Decreasing too much the number of particles does not permit to finish the simulation;
- Adding the unconsolidated zone to the weight equation (weighted mean facies match (Eq. (4.28))) does not improve the well facies match at the end of the simulation compared to the use of the sole consolidated layer. However, considering the unconsolidated zone allows the simulation to continue instead of stopping due to null weights.

The best combination for the factor of the weight equation are $\alpha = 0.8, \beta = 0.2$ and $\gamma = 0$, $\mu = 100$, resampling criterion Min and $p = 250$ (cf. Appendix B.1). Fig. B.1 show the effect of μ, and the number of particles p while the other parameters are fixed for each realization (for each realization, the simulation is chosen by the criterion FM).

Using $p = 10$ implies that the sequential conditioning method does not always finish the simulations, since all weights can be null at a given step due to the severe Min criterion. That is the reason some numbers over the red box plots are not always 25.

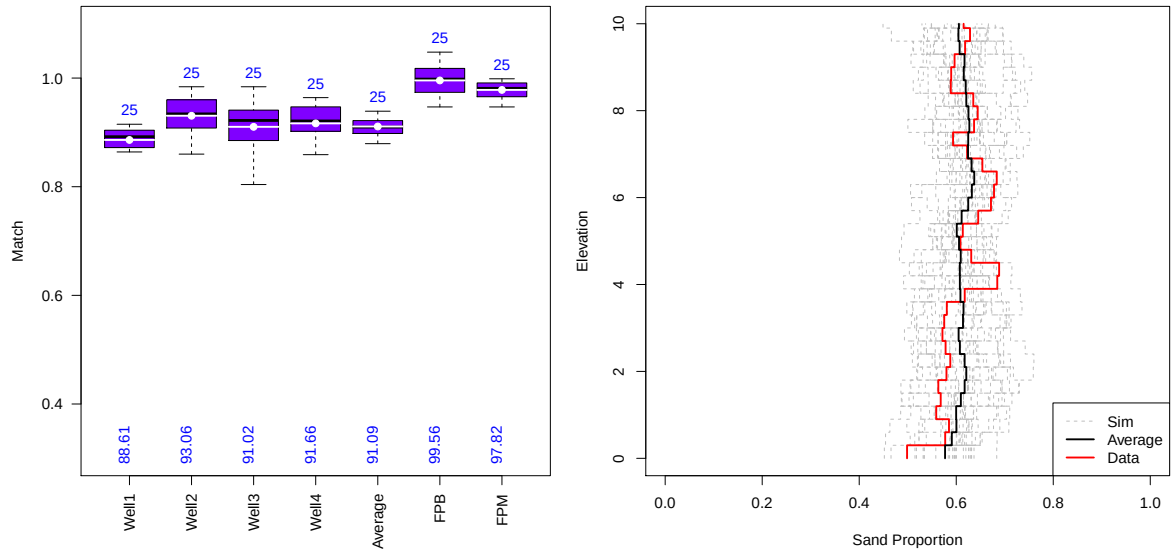


Figure 4.15: Results of the **sequential conditioning method** with the data and conditional simulations grouped in **two-facies wells** for the **synthetic case** from 25 independent realizations. Left, box plots of the facies and grain size class matches for the 4 wells, their mean, the sand proportion balance and the sand proportion match. Right, the sand VPC of reference, of the conditional simulations and their average.

4.3.4.7 Results in the two-facies case

Using the optimal parameters deduced from the previous sensitivity analysis, 25 realizations of the sequential conditioning method have been performed from the two-facies wells only (without considering the sand VPC). Results are shown in Fig. 4.15. The mean well facies match is 91.09% (Eq. (4.9)), the sand proportion balance (Eq. (4.18)) is, on average, 99.56% . It means that 0.44% more sand is deposited in the simulations in average. The sand proportion match (Eq. (4.17)) is 97.82% in average.

The three candidates are chosen among the realizations having the optimal parameters using the three criteria FM, FPM and R on the top left of Fig. 4.16. The candidate FM is then selected and some section views are shown in the same figure. Additionally, a $3D$ view is plotted in Fig. 4.17. Note that the three candidates show the same facies wells (from bottom to top) until the elevation $z = 6m$ (where the candidate R differs for the well 3). Indeed, the Min criterion is very strict. This leads to discard a high number of particles at each resampling step. Consequently, at the end of the algorithm, all proposed particles may share the same evolution history (cf. Appendix B.1)

4.3.5 Simulation with the detailed facies model

The dynamic and sequential conditioning methods are applied to generate Flumy conditional simulations. We are now considering all facies (or called thirteen-facies) ($Nb = 13$) in the conditioning wells (cf. Fig. 4.8), for the calculation of criteria at each step and for the final selection among the realizations. Additionally, the grain size match is analyzed as a by-product of the conditioning results.

4.3.5.1 Results for dynamic conditioning method

For the dynamic conditioning, 250 realizations are computed. Fig. 4.18 shows: the box plots of well matches (facies and grain size) for the 250 realizations and the well data with the three realizations chosen using the three criteria FM, FPM and R. Fig. 4.19 and Fig. 4.20 show sections of a realization when selecting the simulation FM.

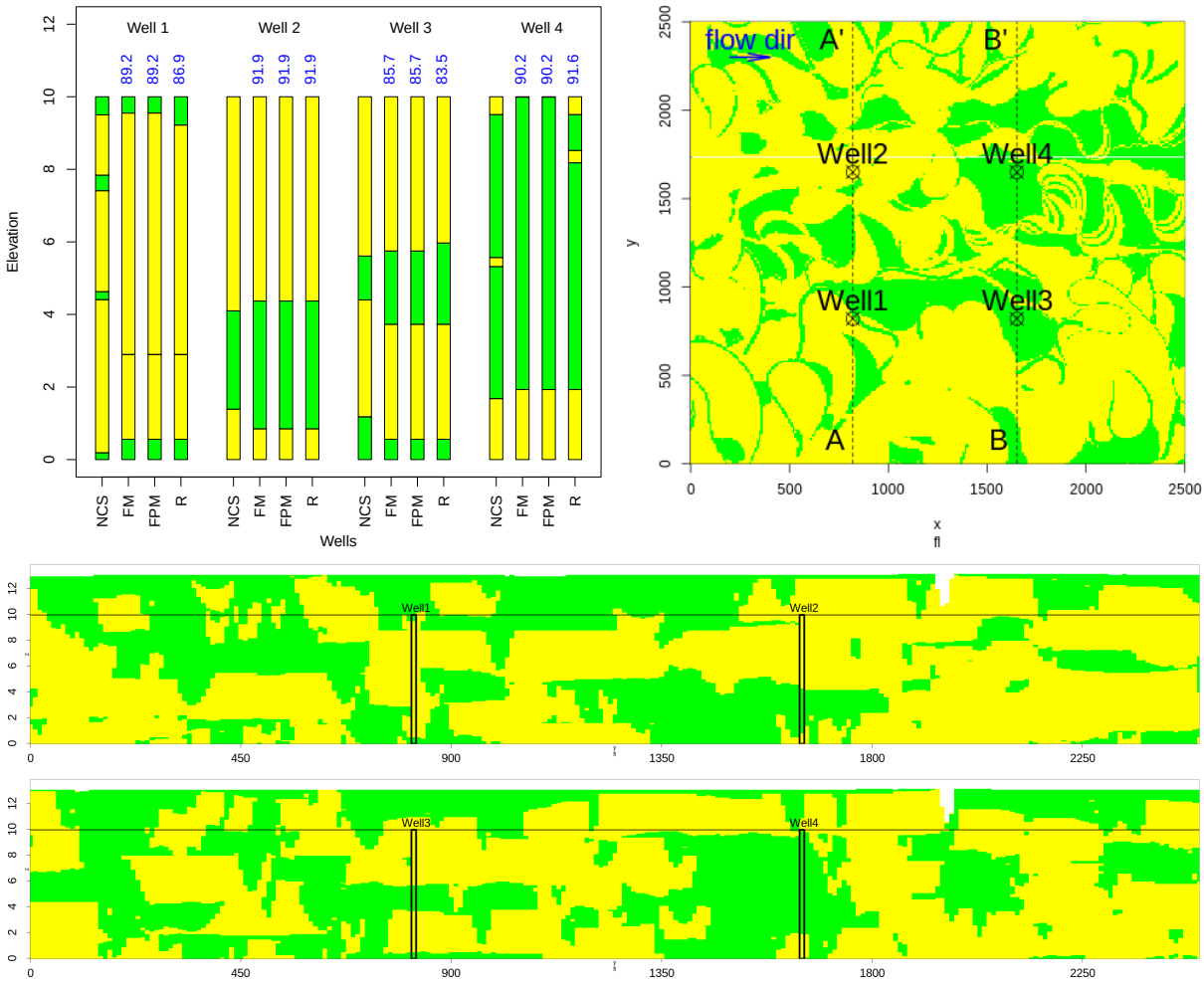


Figure 4.16: From one realization of the **sequential conditioning method** with **two-facies wells** for the **synthetic case**, three candidates are selected using the three criteria FM, FPM and R shown on the top left. Top right, the horizontal slice (z=5m) of the Flumy simulation for the candidate FM. In the middle and on the bottom, the section $A - A'$, and $B - B'$, respectively.

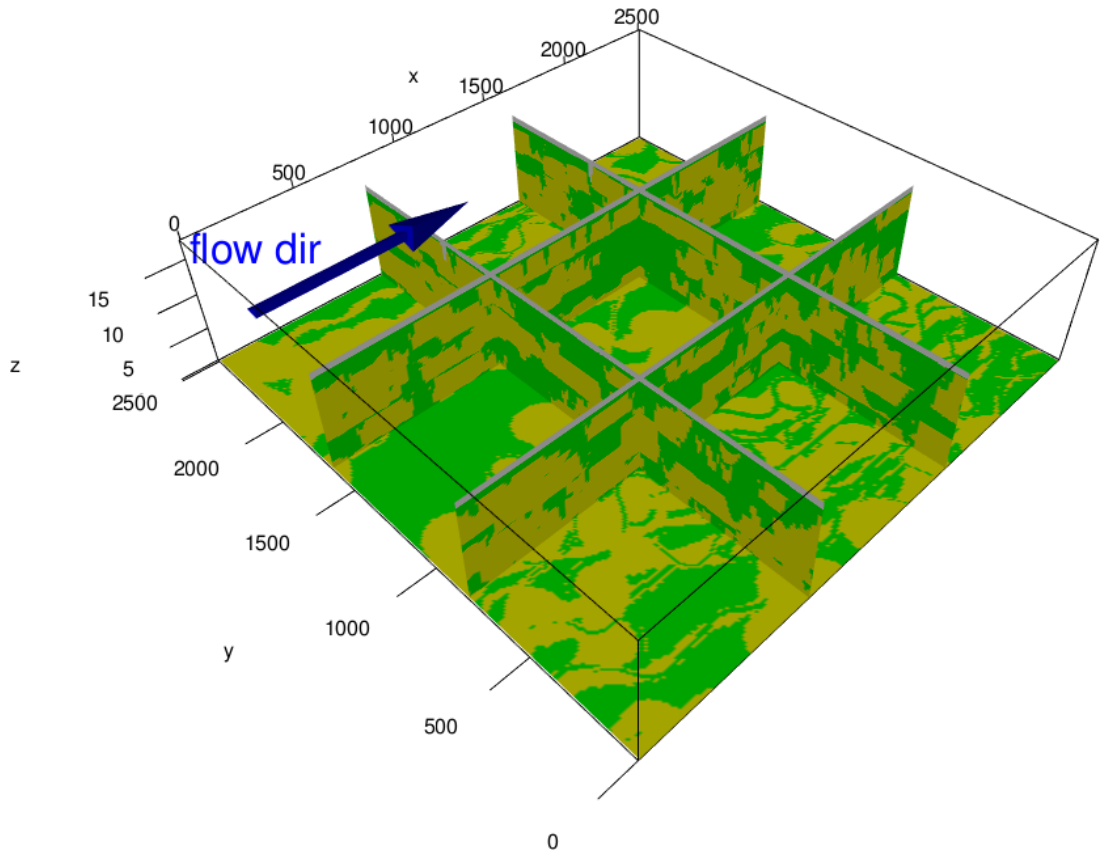


Figure 4.17: $3D$ view of the candidate FM using the **sequential conditioning method** with **two-facies wells** for the **synthetic case**.

The well mean facies match is 57.77%. This result is lower than the two-facies case. While more facies are considered and compared (data vs simulations), it is less probable to get the same facies at the same point. The sand proportion balance is 101.63%. It indicates that, on average, the simulations deposit 1.63% less sand. The sand proportion match shows a match of 97.24%.

It is possible to study the grain size class match in the domain using Eqs. (4.10) and (4.11). The results are shown in the same Fig. 4.18 for the same simulations. On average, the grain size class match is 79.54% .

4.3.5.2 Results for sequential conditioning method

On the other hand, 25 independent realizations for the particle filtering method are performed considering the optimal parameters obtained from the sensitivity analysis in the two-facies case. The results of the matches are shown in the Figs. 4.21. From one realization of the particle filtering method, a reservoir simulation is picked up using criteria FM shown in Fig. 4.22 and Fig. 4.23. One can notice that the selected candidates share the same evolution history than in the two-facies case (see Sect 4.3.4.7). The well facies matches and the mean facies match are lower than the two-facies case, but their values are higher than the dynamic conditioning method. On average, 75.09% of the facies are honored. As in the two-facies wells, the candidates for the same realization tend to have a common history. The facies proportion balance is 100.15% . It means that, on average, 0.15% more sand is deposited by the simulations. The facies proportion match, on average, is 96.64% . It is possible to study the grain size class match, as in the dynamic case, using Eqs. (4.10) and (4.10). The results are shown in the same Fig. 4.21 selecting the same simulations for the well facies match. On average, the grain size match is 85.18% , which is higher than in the dynamic conditioning method.

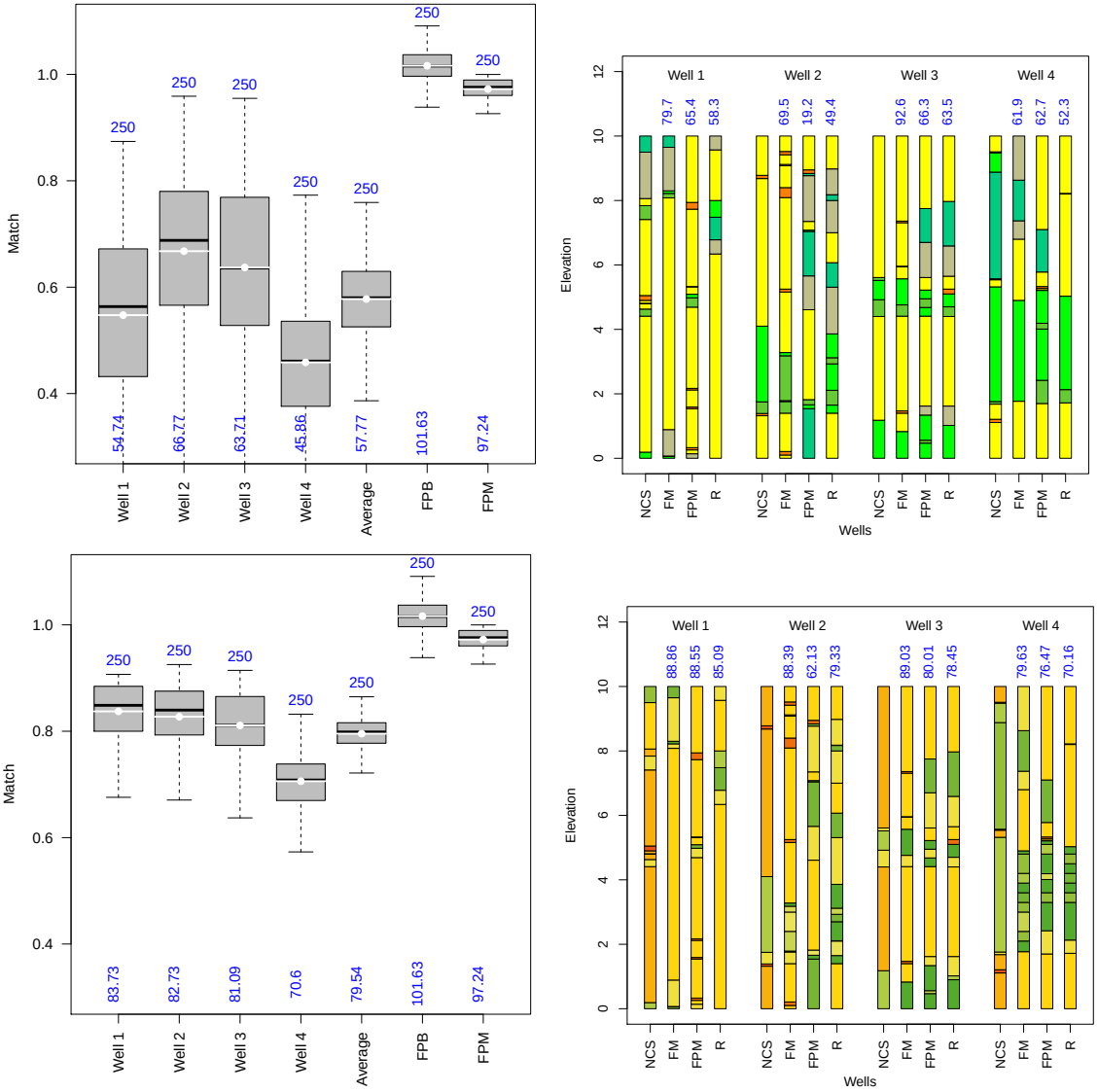


Figure 4.18: Results of the **dynamic conditioning method** with the data and conditional simulations divided into **thirteen-facies wells** for the **synthetic case** from 250 independent realizations. Left, box plots of the facies and grain size class matches for the 4 wells, their mean, the sand proportion balance and the sand proportion match. Right, three simulations are taken with the three criteria FM, FPM and R. Top, 13 facies wells, on the bottom, grain size class wells.

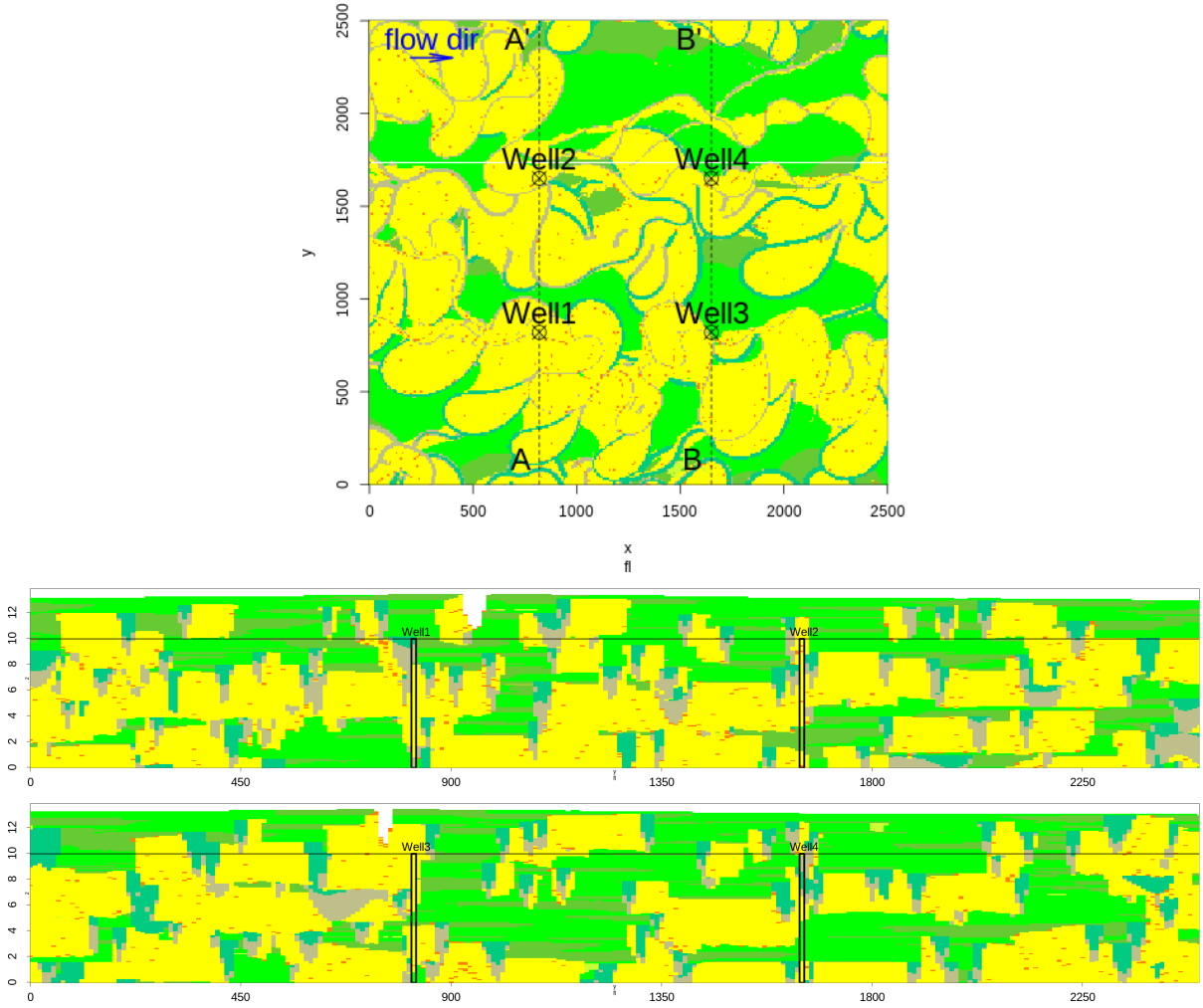


Figure 4.19: Top, the horizontal slice (z=5m) of the Flumy simulation for the simulation FM using the **dynamic conditioning method** with **thirteen-facies wells** for the **synthetic case**. In the middle and on the bottom, the section $A - A'$, and $B - B'$, respectively.

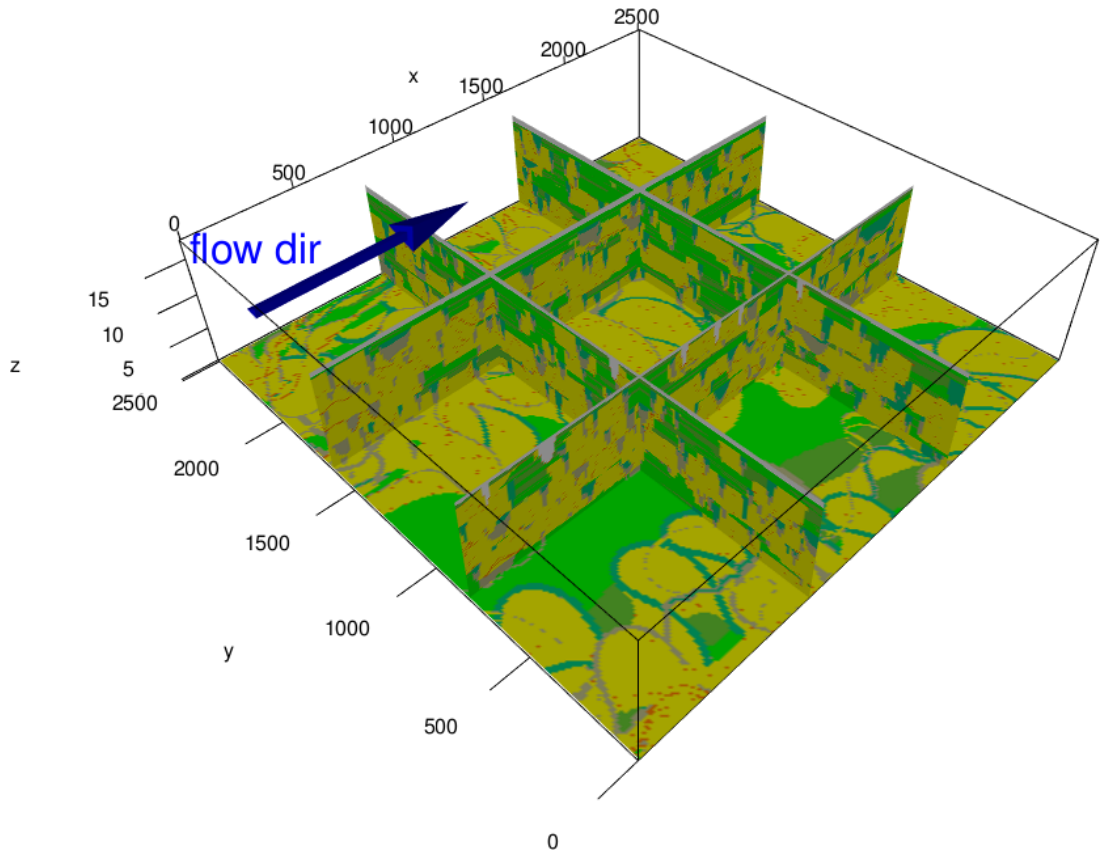


Figure 4.20: $3D$ view of a realization for the FM criterion using the **dynamic conditioning method** with **thirteen-facies wells** for the **synthetic case**.

4.3.6 Discussion for the synthetic case

The dynamic conditioning method, and the sequential conditioning method, have been used to perform conditional simulations for the Flumy process-based model. The first method does a conditional simulation considering well constraints only. The second one, sequential, can perform a conditional simulation considering two types of constraints: wells and sand VPC (although this last constraint seems not to improve the results in our case). The natural question is how do they compare from different perspectives: well matches, sand VPC match, running time, and memory footprint. For the wells constraints, two kinds of information are considered : two facies sand/clay, or thirteen-facies.

The dynamic conditioning method offers good matching to the conditioning well data in the two-facies case. From 250 independent simulations, the mean facies match is 85.15%.

On the other hand, when thirteen-facies are considered, the results get worse. From 250 independent simulations, the mean facies match is 57.77%.

Regarding time performance and memory footprint, the dynamic conditioning method is faster than the sequential alternative, since only one realization is enough to produce a conditional simulation. Moreover, the dynamic conditioning method requires far less memory than the sequential conditioning method. In consequence, to measure the uncertainty of the conditioning methods, it is faster to obtain independent simulations for the dynamic method than the sequential method.

However, there is a major issue regarding the grain size. The dynamic conditioning tends to deposit a channelized facies at wells where sand is required. Unfortunately, one of these channelized facies has a fine grain size (i.e., the Mud Plug used to fill abandoned channels bits (Fig. 4.3)). Conversely, some high grain sized facies (sand) can be deposited in place of mud plug as both facies are considered channelized. As a consequence, an inconsistency can appear

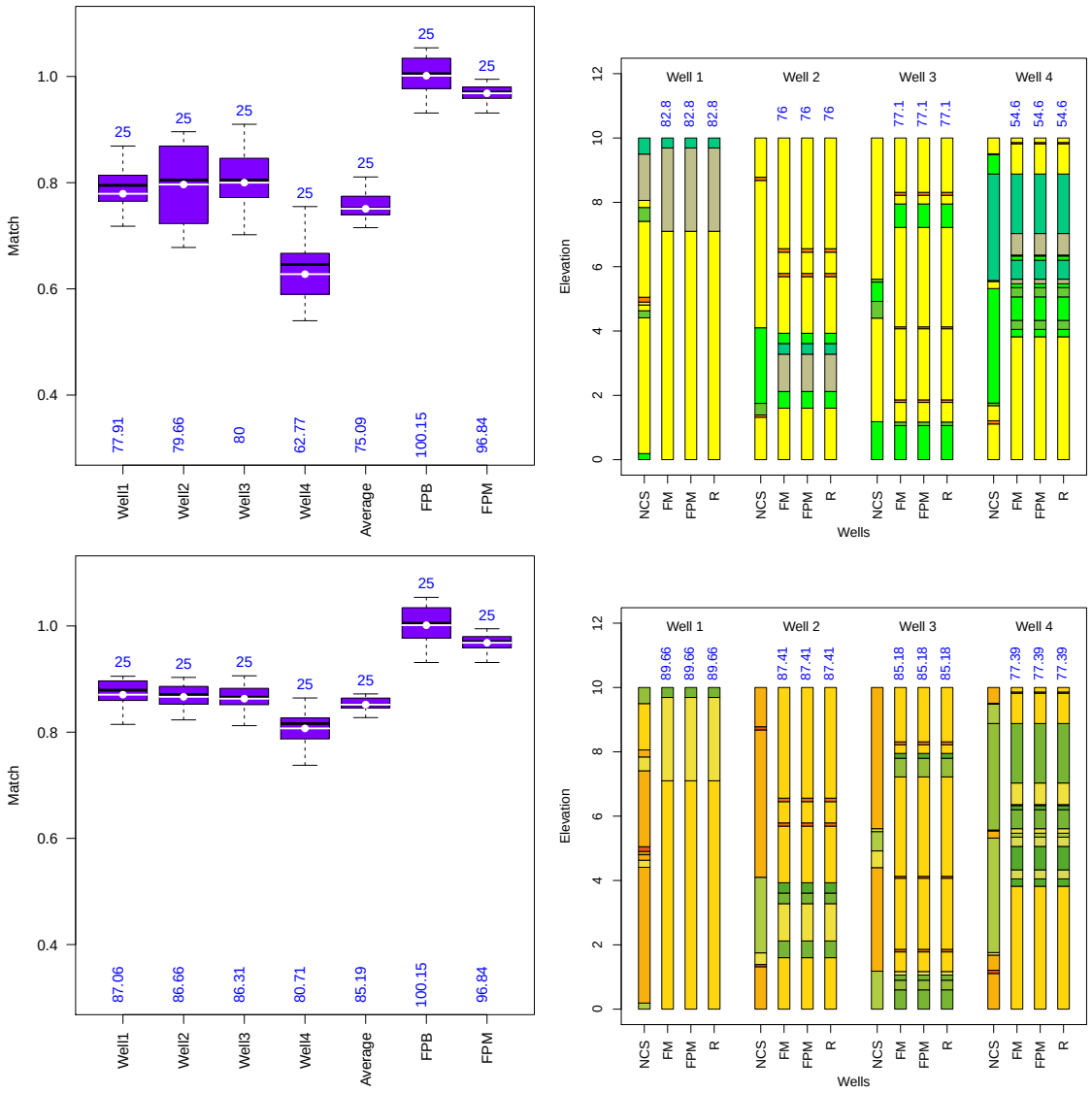


Figure 4.21: Results of the **sequential conditioning method** with the data and conditional simulations divided into **thirteen-facies wells** for the **synthetic case** from 25 independent realizations. Left, box plots of the facies and grain size class matches for the 4 wells, their mean, the sand proportion balance and the sand proportion match. Right, from one realization of the sequential conditioning method, three candidates are selected with the three criteria FM, FPM and R. Top, 13 facies wells, on the bottom, grain size class wells.

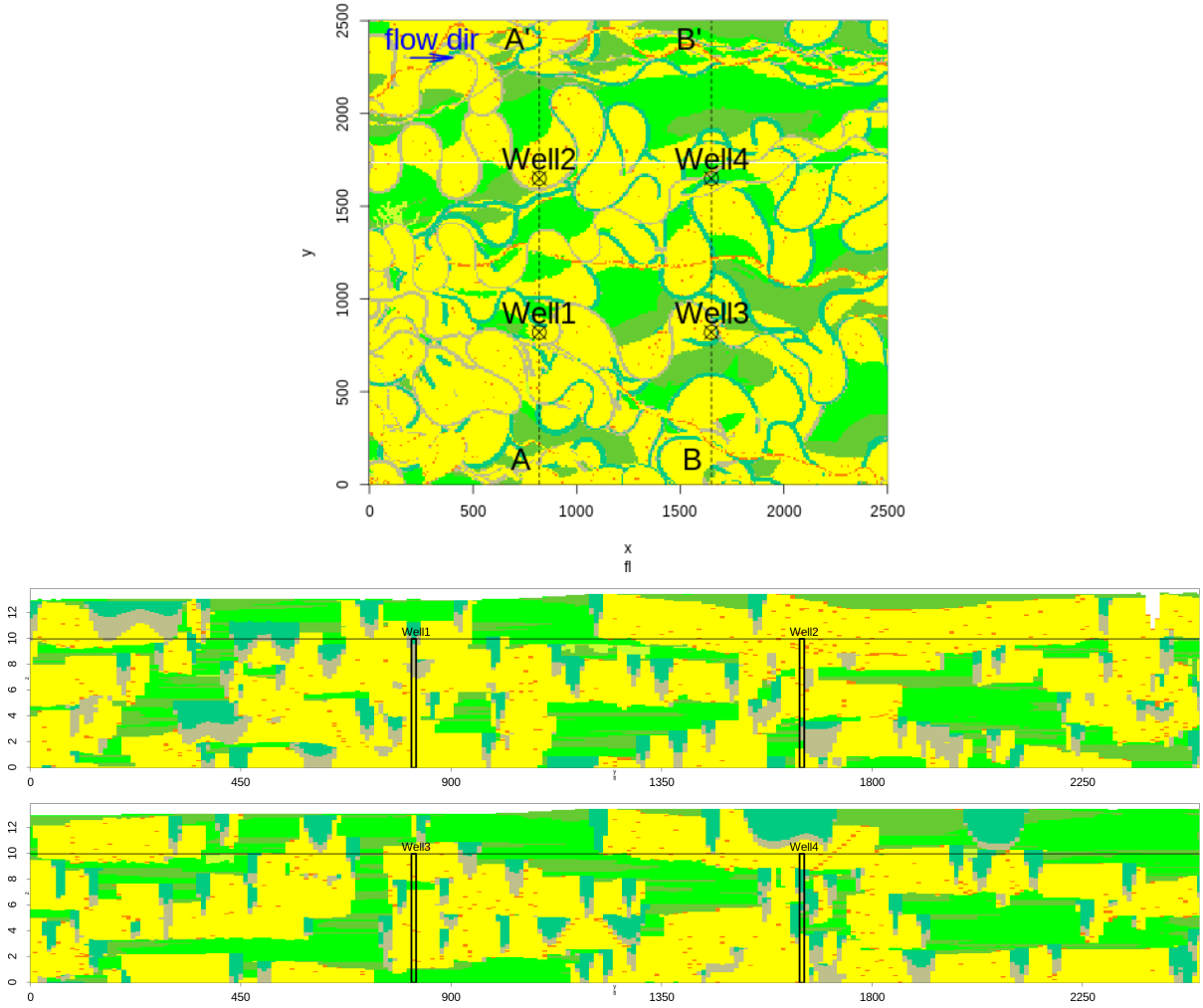


Figure 4.22: Top, the horizontal slice (z=5m) of the Flumy simulation for the particle FM using the **sequential conditioning method** with **thirteen-facies wells** for the **synthetic case**. In the middle and on the bottom, the sections $A - A'$, and $B - B'$, respectively.

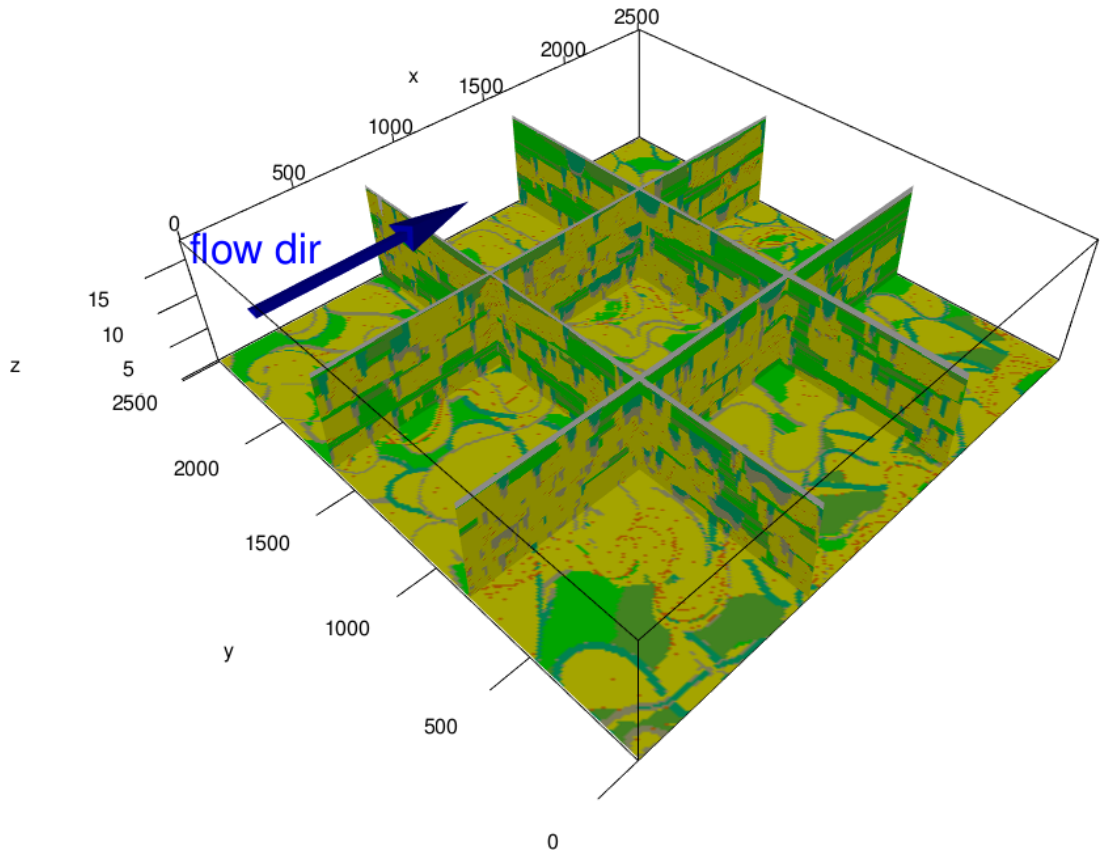


Figure 4.23: $3D$ view of the candidate FM using the **sequential conditioning** with **thirteen-facies wells** for the **synthetic case**.

Table 4.3: For each column: Sequential method: Mean over 25 independent realizations, selecting the particles having the best facies match (FM). Dynamic method: mean over 250 independent realizations. Synthetic case.

Cases	Conditioning Method	Facies Match	Facies Proportion Balance	Facies Proportion Match	Grain Match
$Nb = 2$	Dynamic	85.15%	101.97%	97.24%	
	Sequential	91.09%	99.56%	97.82%	
$Nb = 13$	Dynamic	57.77%	101.63%	97.24%	79.54%
	Sequential	75.09%	100.15%	96.64%	85.18%

in the reservoir, adding clays where, possibly, there were sands (or conversely). However, the geological context (proximity of the channel) is honored. Moreover, the match is computed supposing that the wells are exactly located at grid nodes, which in practice is, in general, not the case. The mean grain size match for the dynamic conditioning method is 79.54% .

Using the two-facies classification, considering 25 independent simulations of the sequential conditioning method, the mean well facies match is 91.09% . On the other hand, using the thirteen-facies classification, considering 25 independent simulations, the mean well facies match is 75.09% . In both cases, the sequential conditioning method outperforms the dynamic version on average.

In terms of grain size, the facies deposited are coherent with the grain size class of the conditioning data. The mean well grain size class match for the dynamic conditioning method is 79.54% , while in the sequential version, it is 85.18% .

Table 4.3 gives a summary of the matches for the synthetic case. Moreover, it is possible to incorporate directly grain size constraints in the sequential method. The weight equation from

Eq. (4.31) can be modified by adding Eqs. (4.10) and (4.11) in order to evaluate this conditioning information at each step.

Another major issue is that the dynamic conditioning method modifies the behavior of the simulated processes (increasing/decreasing locally the migration rate, blocking temporarily the aggradation or tossing the new channel paths not necessarily in the deepest part of the topography). This means that the dynamic conditioning method can create some distortions in the shape of the sand bodies or in the sand distribution compared to non-conditional simulations. Huge work has been made to minimize these undesirable effects (Bubnova, 2018). Thus, by comparing the sections of the reference simulation (Fig. 4.6) and the one from the corresponding dynamic conditioning simulation (Fig. 4.19), no visual difference is noticed. However, we know that by adding more wells, the dynamic conditioning tends to increase lateral extensions of the point bars and produce some sand clusters in some parts of the domain.

An advantage of the sequential conditioning method, over the dynamic one, is that it only runs the non-conditional model. Consequently, the reservoirs produced by the sequential conditioning method have the same properties as the non-conditional model, without any artifact or distortion. Finally, particle filtering also offers p candidates as equiprobable conditional simulations for each realization. This permits to select several plausible scenarios (non-independent) without having to increase the number of realizations.

However, some drawbacks exist in the sequential conditioning method. To get a good match with the conditioning data, a significant number of particles are needed to explore the simulation space. Moreover, while more data constraints are added, more particles are needed. This could lead to increase memory and processing time issues.

4.4 Application of the Flumy model to the Loranca Basin (Spain)

The Loranca basin is located in the central part of Spain (Fig. 4.24), within the Iberian range. The basin was formed in the Eocene to late Oligocene, and contains continental deposits (Díaz-Molina et al., 1993). From the late Oligocene to the Early Miocene times, the Loranca Basin was filled by two coalescing fluvial systems, the Tórtola and the Villalba de la Sierra fluvial fans (Díaz Molina et al., 1989). The Lower-Miocene deposits of the Loranca formation are dominated by meandering fluvial deposits in the Huete area, corresponding to the distal part of the system. Along the flanks of the Huete anticline, the sedimentary succession consists of fluvial plain deposits, including meander loops, abandoned meander channels, channel fills, crevasse splay and levees (Daams et al., 1996). The fluvial deposit outcrops are an exceptional and useful characteristics of the Loranca Basin. These features allow researchers to take samples and construct geological sections. The paleo-flow direction towards the NW remained stable along the studied area. In this work, a rotation of the system is performed to direct the fluvial channel in direction S-N. Eight pseudo-wells have been created from the facies observed on the outcrops. From these wells, a vertical sand proportion curve has been calculated in order to verify the existence of sedimentary units. No section has been created in areas where there were no point bars. In order to compensate, an additional virtual well full of clay facies has been added in one specific area having no sand. This additional well is not considered for the sand VPC. This *repulsive* well will prevent conditional simulations of Flumy from depositing sand in this area.

Flumy's domain parameters are defined as:

- OX, OY: 0×0m;
- NX and NY: 371×451
- DX and DY : 10×10m.

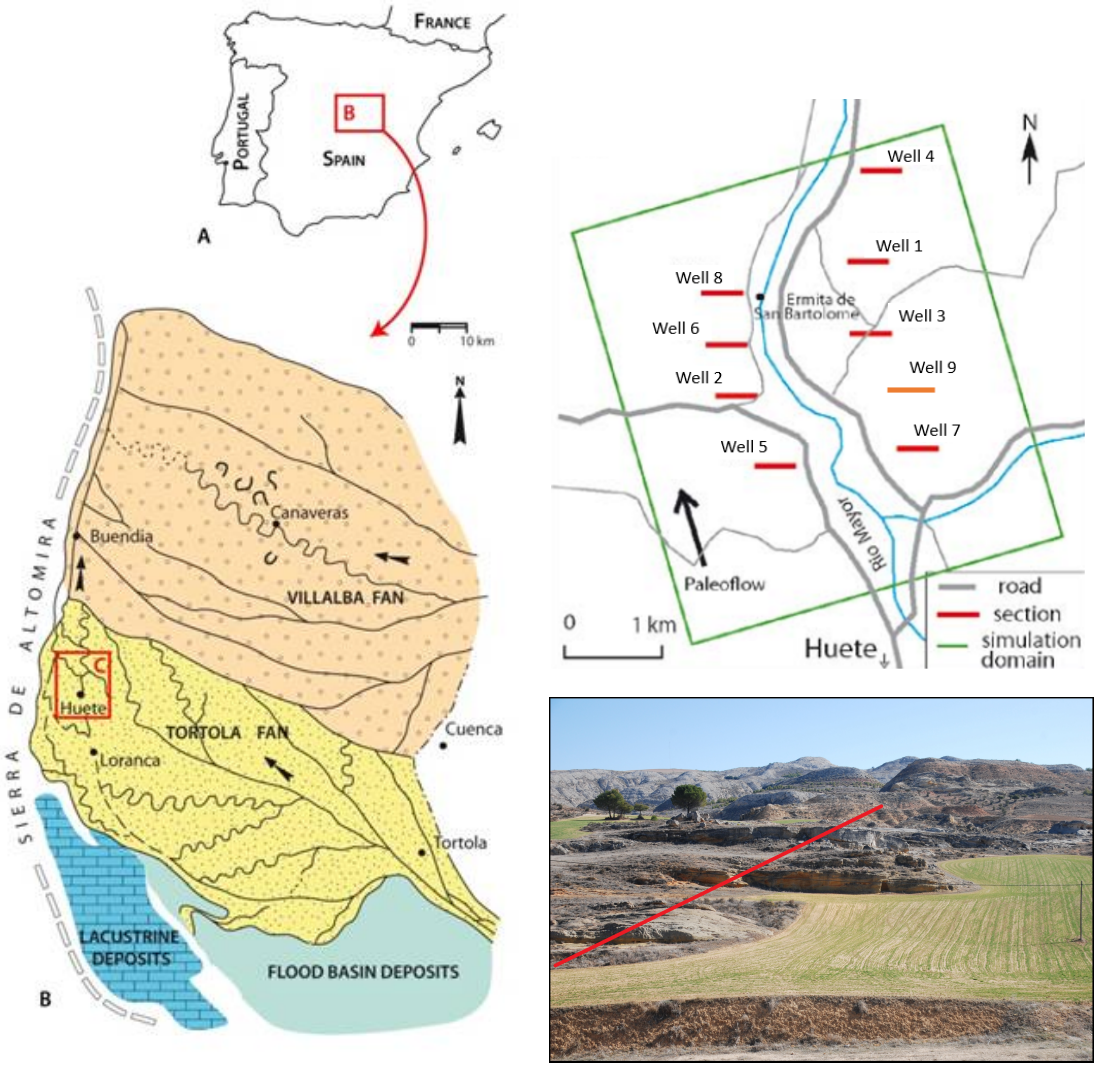


Figure 4.24: Left, location of the Loranca Basin (C) in the Tortola fan, near Huete village (in Spain). Top right, the simulated domain with 9 wells. A clay well 9 is added artificially (in orange). Bottom right, global view of the Peña de San Juan section (in red) corresponding to well 7.

Table 4.4: Maximum channel depth H_{max} and N/G for each Sedimentary Unit (S.U.) of the Loranca basin.

Sedimentary Unit	**Elevation Limits** m	H_{max} m	**N/G** %
S.U.3	$[58, 90]$	2.29	27
S.U.2	$[34, 58]$	2.98	52
S.U.1	$[0, 34]$	2.39	28

The domain size is then 3710 x 4510m. The reservoir vertical limit is 90m (starting from 0). There are three different sedimentary units in the reservoir. These units have been obtained by applying the Geostatistical Hierarchical Clustering (GHC) (Bubnova et al., 2020) to the eight pseudo-wells. The maximum depth of the channel into these units has been inferred from the pseudo-wells following the procedure from Cojan et al. (2012). The rest of the Flumy parameters have been retained from Weill et al. (2013) (more details of the Flumy parameters are given in Table B.1). The conditioning wells, and the well sand VPC are shown in Fig. 4.25. The sand VPC is obtained from the sand proportion at wells weighted by their contribution in the different layers.

4.4.1 Simulation with a simplified sand/shale facies model

4.4.1.1 Results for dynamic conditioning method

100 independent realizations using the dynamic conditioning method are performed considering the 9 available two-facies wells sand-clays ($Nb = 2$). The well facies match, their weighted mean facies match, the sand proportion balance and the sand proportion match are shown in Fig. 4.26. The sand VPC from well data, and from the conditional simulations, are shown in the same figure. The value of the mean facies match of the 100 independent realizations is 87.69%. The sand facies balance shows that, on average, is 98.53%. This means that simulations deposit on average 1.47% more sand than the data. The sand proportion match shows, on average, a 98.52% of match.

Like in the synthetic case, the simulations FM, FPM and R are selected among the 100 independent realizations. Their match results by well are shown on top left in Fig. 4.27. The simulation FM is chosen, and different sections are shown in the same figure. A $3D$ view is shown in Fig. 4.28.

4.4.1.2 Results for sequential conditioning method

The sequential conditioning method is applied using the optimal parameters as for the synthetic case. However, during its implementation and execution, an important issue arises. The Min criterion (Eq.(4.24)) produces null weights when (usually) more wells are incorporated in the calculi. This phenomenon occurs in the middle of the reservoir simulation, when more wells are simultaneously considered. As a consequence, the algorithm is stopped. In order to avoid this problem, an alternative criterion is adopted. When all weights are null for a certain step i, the Average criterion is adopted (Eqs. (4.23) and (4.29)) for the consolidated layer and unconsolidated zone, respectively. This relaxation allows continuing the algorithm. It appears that the Average criterion is called less than 5% of the time. A number of 6 independent realizations are computed. Fig. 4.29 shows the box plots of well facies match, their weighted mean facies match, the facies proportion balance and the facies proportion match. The value of the mean facies match of 6 independent realizations is 77.93%, lower than the dynamic case. On average, the sand facies balance is 97.98% showing that simulations have 2.02% more sand than the sand data. The sand proportion match is, on average, 97.98%.

For a particular realization of the particle filtering method, the three candidates are selected with the three criteria FM, FPM and R shown in Fig. 4.30. The sections view of FM are

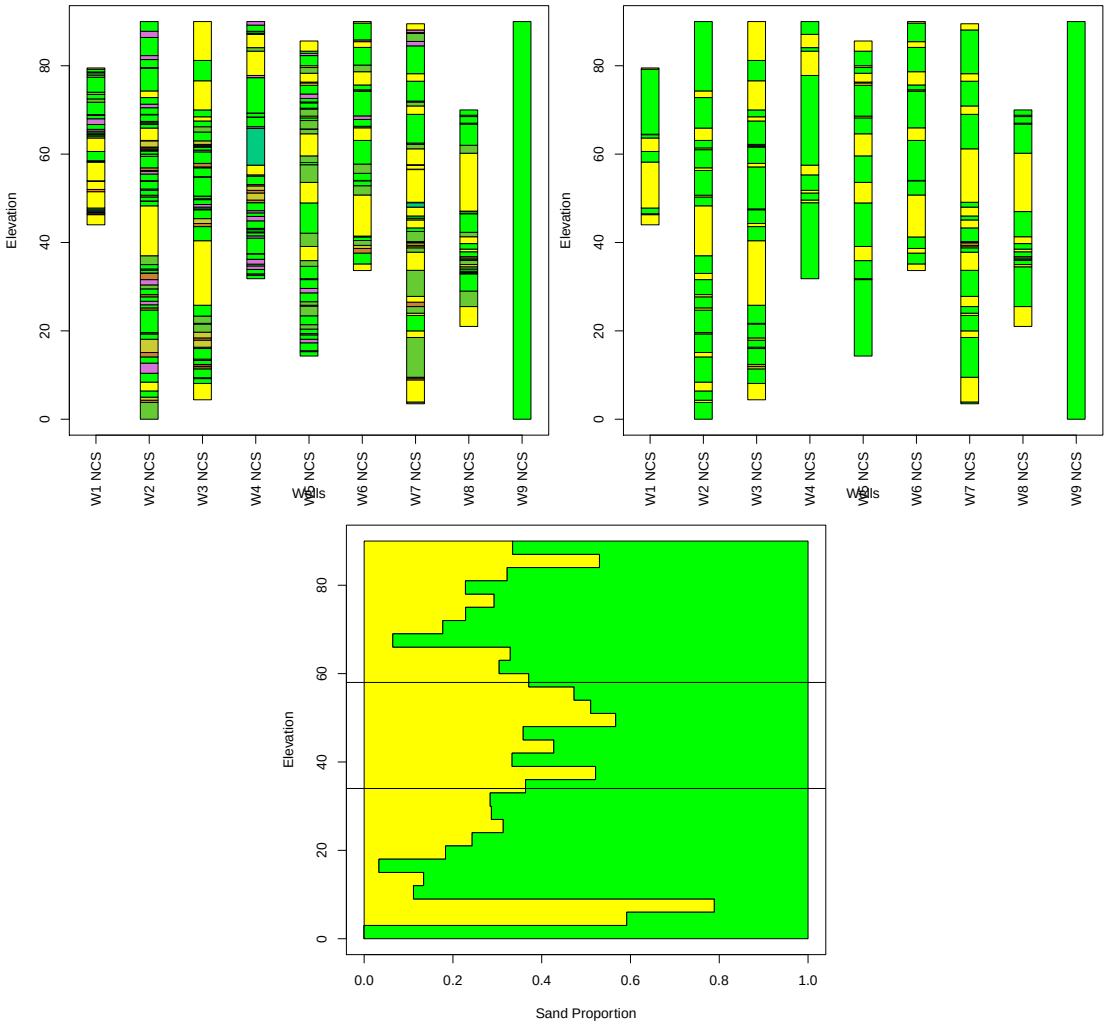


Figure 4.25: Top-left, the thirteen-facies wells and on the top-right, the two-facies wells. W9 is the virtual well added to prevent the sampling bias. Bottom, the sand VPC (computed without W9) of Loranca case study. Horizontal lines in the sand VPC represent the sedimentary unit limits.

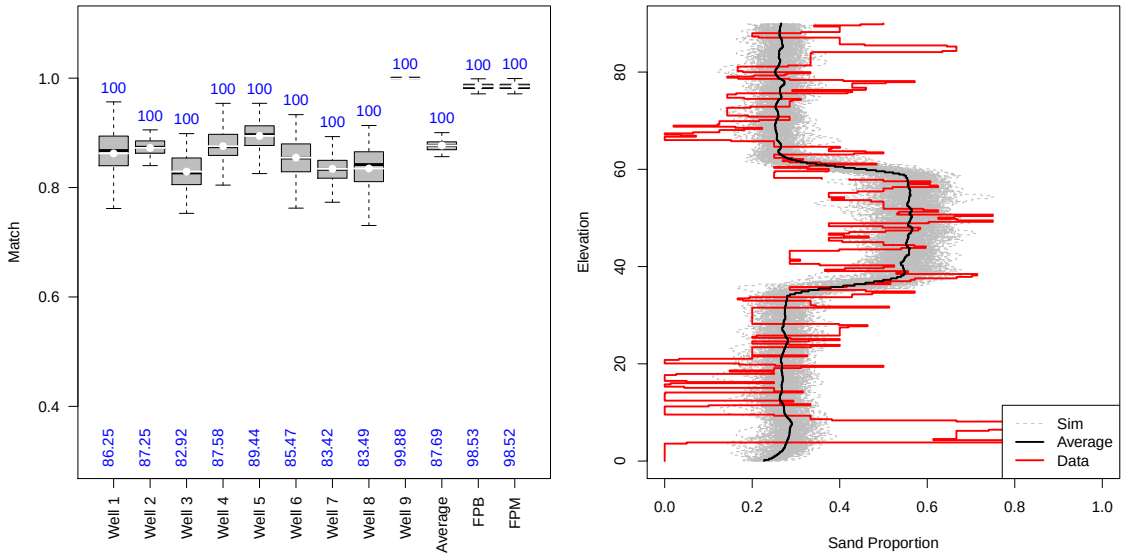


Figure 4.26: Results of the **dynamic conditioning method** with data and conditional simulations grouped in **two-facies wells** for the **Loranca case study**. Left, box plots of the facies and grain size class matches for the 9 wells, their mean, the sand proportion balance and the sand proportion match. Right, the sand VPC of the well data, of the conditional simulations and their average. Each layer of the sand VPC has the same size as the layers for the sequential method.

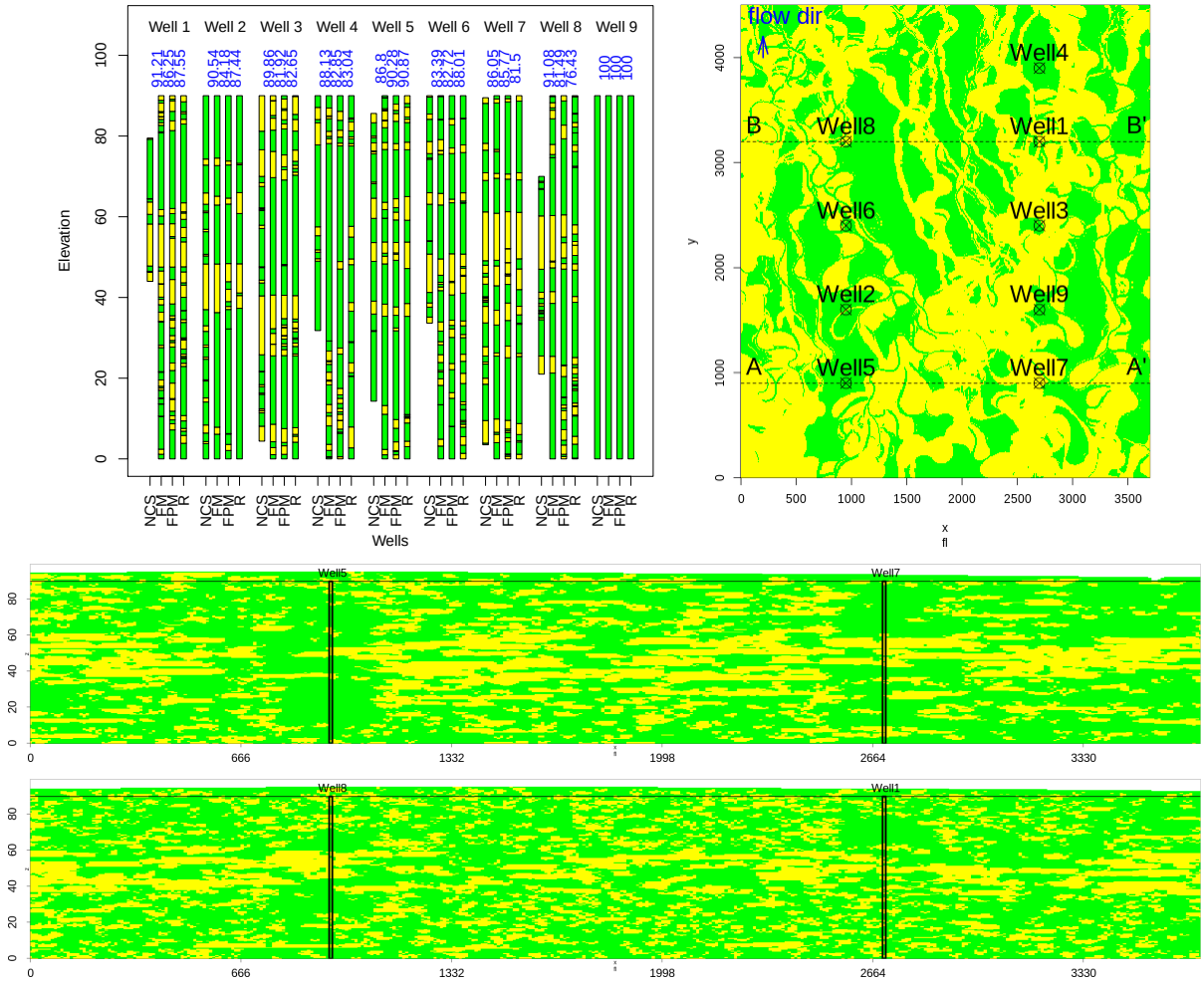


Figure 4.27: From 100 realizations of the **dynamic conditioning method** with **two-facies wells** for the **Loranca case study**, three simulations are selected using the three criteria FM, FPM and R are shown on the top left. Top right, the horizontal slice (z=45m) within the second unit of the Flumy simulation. In the middle and on the bottom, the sections $A - A'$, and $B - B'$, respectively.

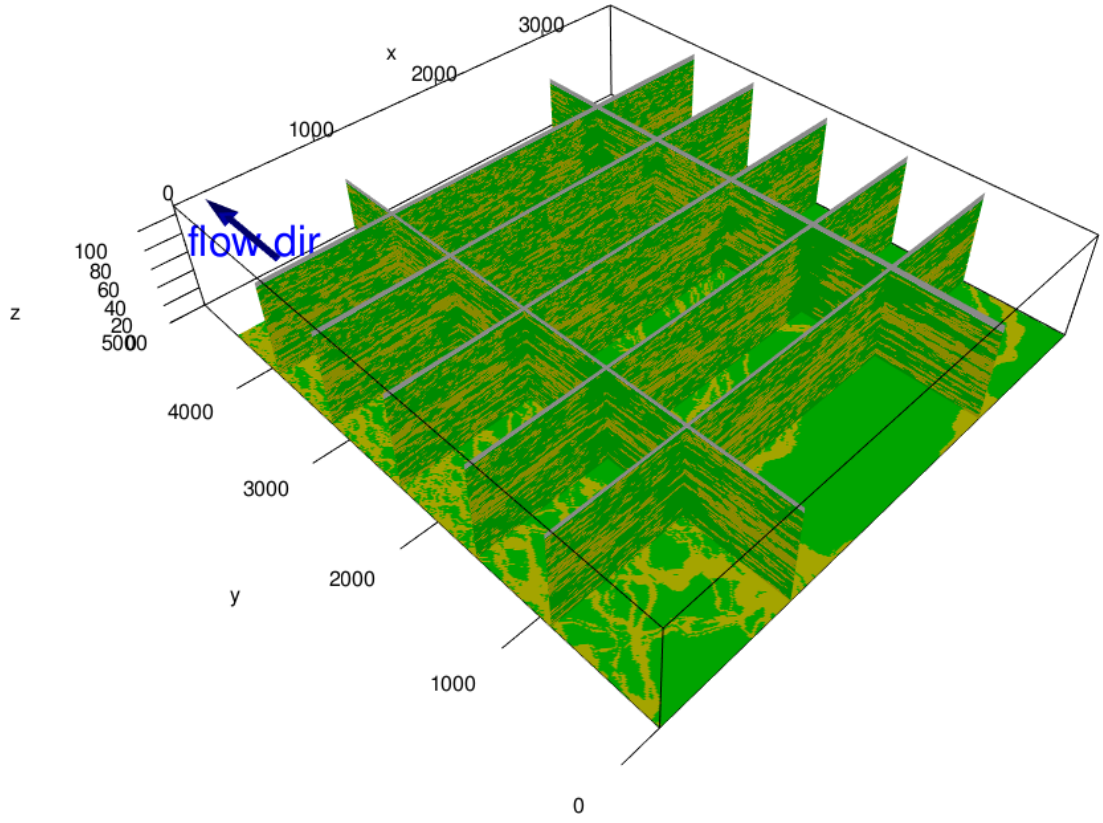


Figure 4.28: $3D$ view of a realization for the FM criterion using the **dynamic conditioning method** with **two-facies wells** for the **Loranca case study**.

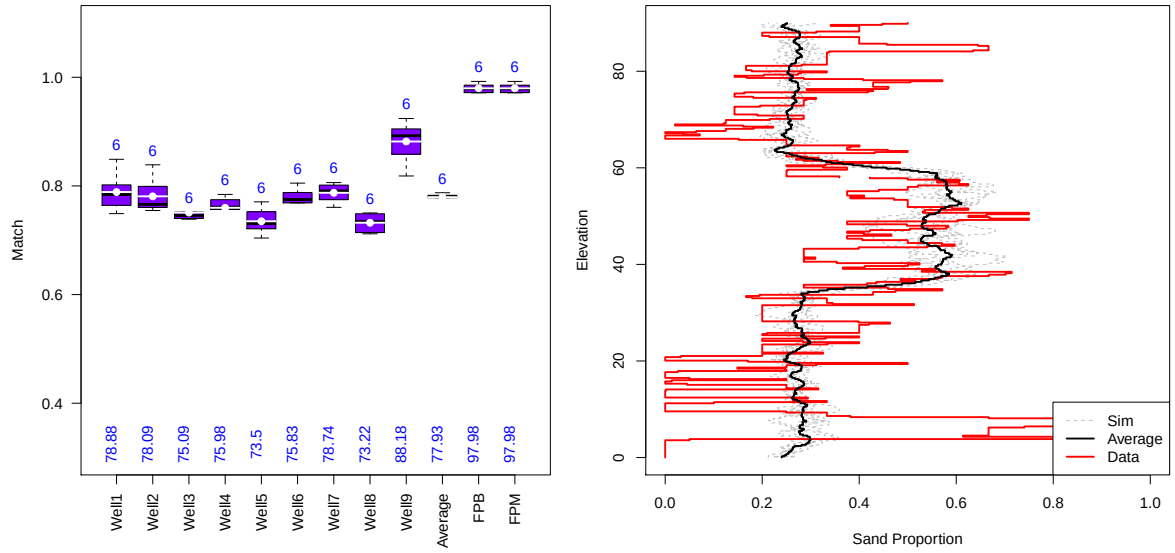


Figure 4.29: Results of the **sequential conditioning method** with the data and conditional simulations grouped in **two-facies wells** for the **Loranca case study** from 6 independent realizations. On the left, box plots of the facies matches for the 4 wells, their mean, the sand proportion balance and the sand proportion match. Right, the sand VPC of well data, of the conditional simulations and their average.

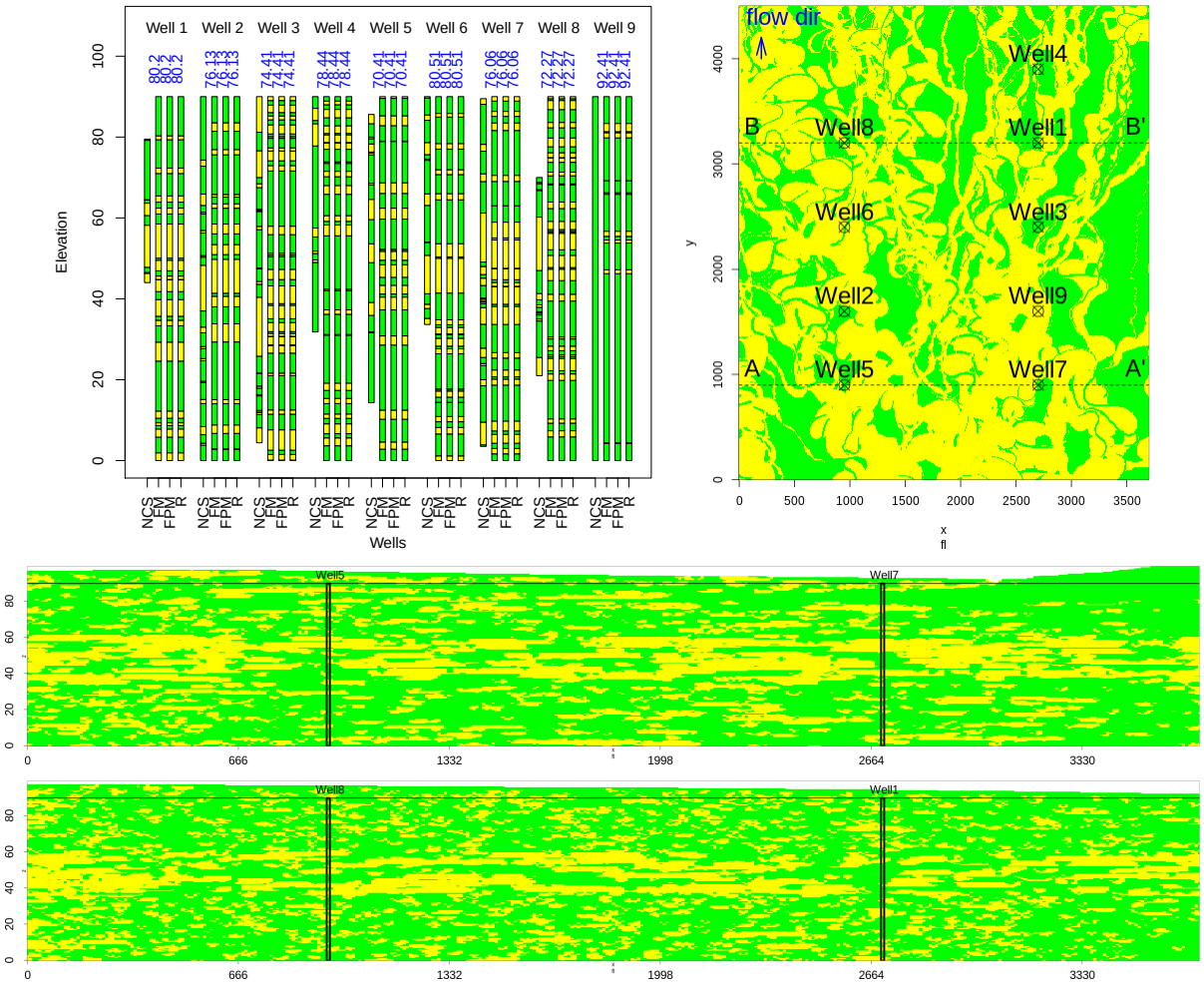


Figure 4.30: From one realization of the **sequential conditioning method** with **two-facies wells** for the **Loranca case study**, three candidates are selected using the three criteria FM, FPM and R are shown on the top left. Top right, the horizontal slice (z=45m) within the second unit of the Flumy simulation for the candidate FM. In the middle and on the bottom, the cross sections $A - A'$, and $B - B'$, respectively.

shown in the same figure. Finally, a $3D$ view is shown in Fig. 4.31.

4.4.2 Simulation with the detailed facies model

4.4.2.1 Results for dynamic conditioning method

The dynamic and sequential conditioning methods are applied considering all Flumy facies ($Nb = 13$), for the well data and for the criteria calculation from the conditional simulations.

For the dynamic conditioning, 100 realizations are computed. Fig. 4.32 shows the box plots of well facies match, their weighted mean facies match, the facies proportion balance and the facies proportion match. As in the dynamic conditioning method of the synthetic case, the weighted mean facies match is lower than the two-facies case. On average, the facies match is 59.21%. The facies proportion balance is, on average, 98.34%. This means that, on average, the simulations deposit 1.66% more sand than data. The facies proportion match is, on average, 98.34%.

Fig. 4.32 shows three simulations of Loranca basin using the three criteria FM, FPM and R. The simulation FM is represented in section views and in a $3D$ view in the Figs. 4.33 and 4.34.

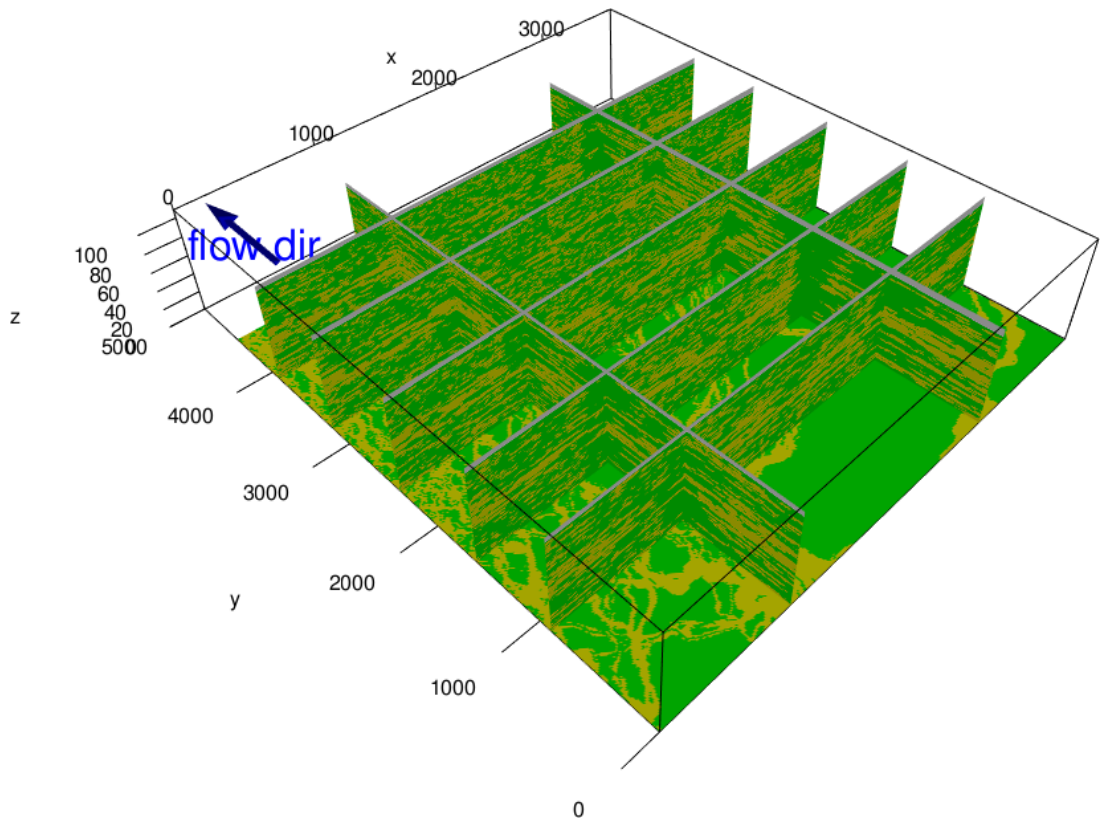


Figure 4.31: $3D$ view of the candidate FM using the **sequential conditioning method** with **two-facies wells** for the **Loranca case study**.

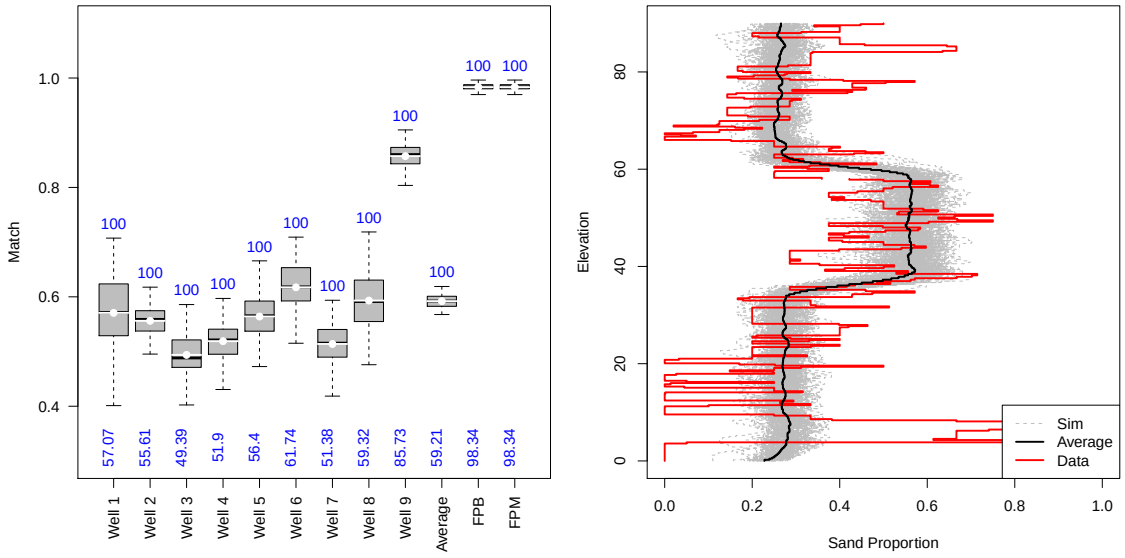


Figure 4.32: Results of the **dynamic conditioning method** with the data and conditional simulation divided into **thirteen-facies wells** for the **Loranca case study** from 6 independent realizations. Left, box plots of the facies and grain size class matches for the 9 wells, their mean, the sand proportion balance and the sand proportion match. Right, the sand VPC of the well data, of the conditional simulations and their average.

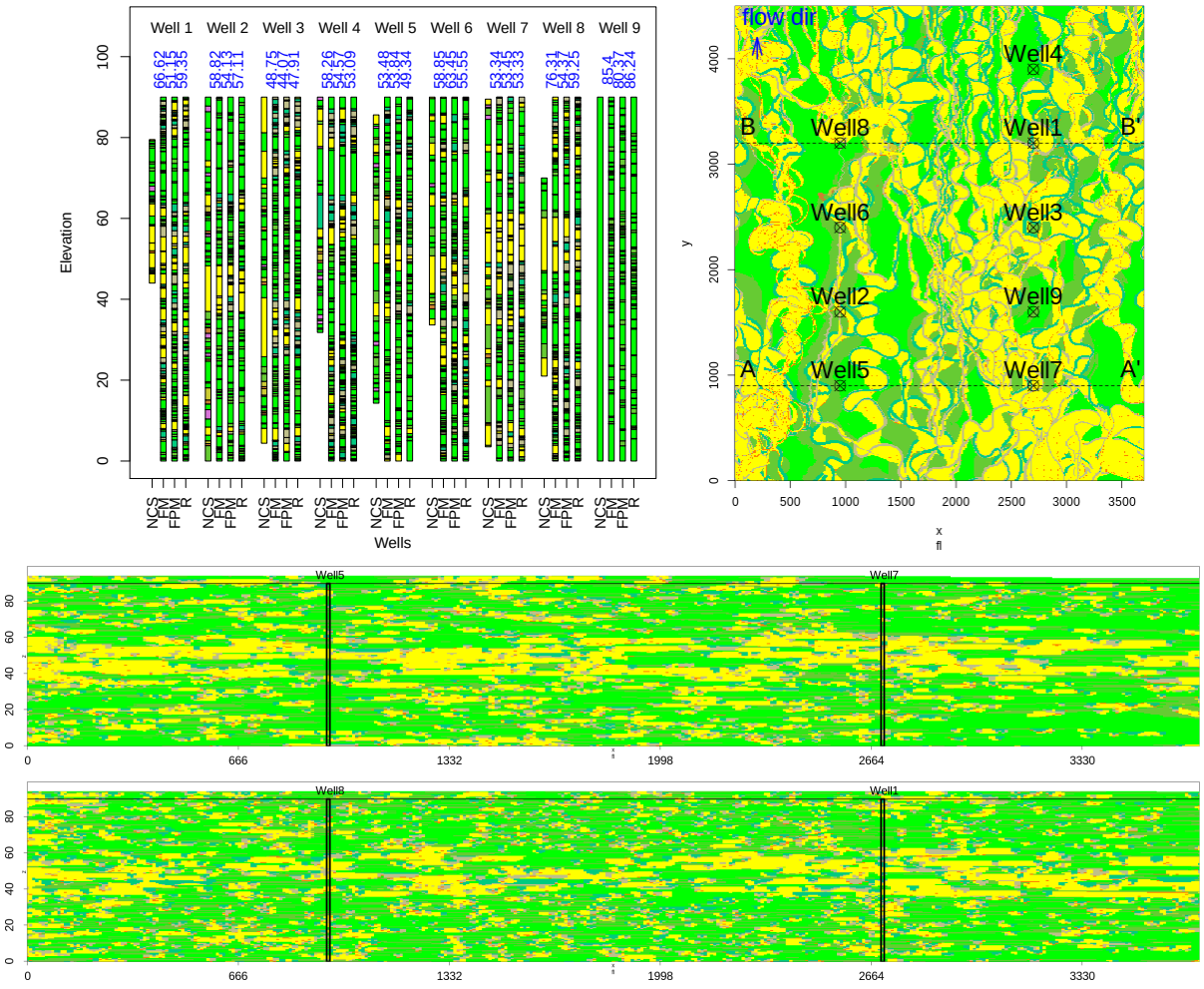


Figure 4.33: From 100 realizations of the **dynamic conditioning method** with **thirteen-facies wells** for the **Loranca case study**, three simulations are selected using the three criteria FM, FPM and R are shown on the top left. Top right, the horizontal slice (z=45m) within the second unit of the Flumy simulation for the simulation FM. In the middle and on the bottom, the sections $A - A'$, and $B - B'$, respectively.

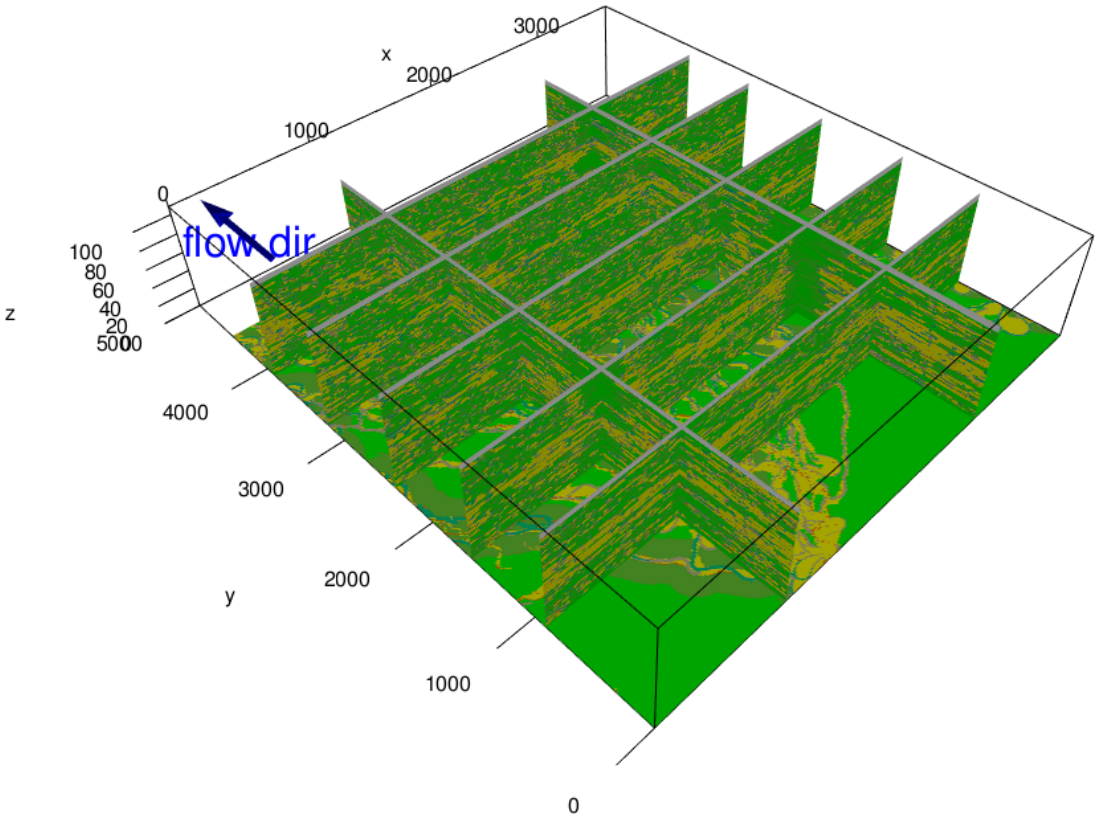


Figure 4.34: $3D$ view of the simulation FM using the **dynamic conditioning** with **thirteen-facies wells** for the **Loranca case study**.

Table 4.5: For each column: Sequential method: Mean over 6 independent realizations, selecting the particles having the best facies match. Dynamic method: mean over 100 independent realizations. Loranca case study.

Cases	Conditioning Method	Facies Match	Facies Proportion Balance	Facies Proportion Match
$Nb = 2$	Dynamic	87.69%	98.53%	98.52%
	Sequential	77.93%	97.98%	97.98%
$Nb = 13$	Dynamic	59.21%	98.34%	98.34%
	Sequential	47.31%	98.42%	98.42%

4.4.2.2 Results for sequential conditioning method

For the sequential conditioning method, 6 realizations are computed. Fig. 4.35 shows the box plots of facies match of the wells, their weighted mean facies match, facies proportion balance and facies proportion match. As in the two-facies case, the Min criterion cannot continue the algorithm, so the Average criterion is used when all weights are null. Therefore, between $20-35\%$ times, the Average criterion is called. Fig. 4.35 shows the box plots of well facies match, their weighted mean facies match, the facies proportion balance and the facies proportion match. As in the sequential conditioning method of the synthetic case, the weighted mean facies match is lower than in the two-facies case. Moreover, the match is lower than in the dynamic conditioning method. On average, the average facies match is 47.31%. The facies proportion balance is, on average, 98.42%. This means that, on average, the simulations deposit 1.58% more sand than data. The facies proportion match is, on average, 98.42%.

Fig. 4.36 shows three candidates using the three criteria FM, FPM and R. The same figure shows three section views for the candidate FM. Fig. 4.37 shows a $3D$ view. Table 4.5 gives a summary of the results for the real case study.

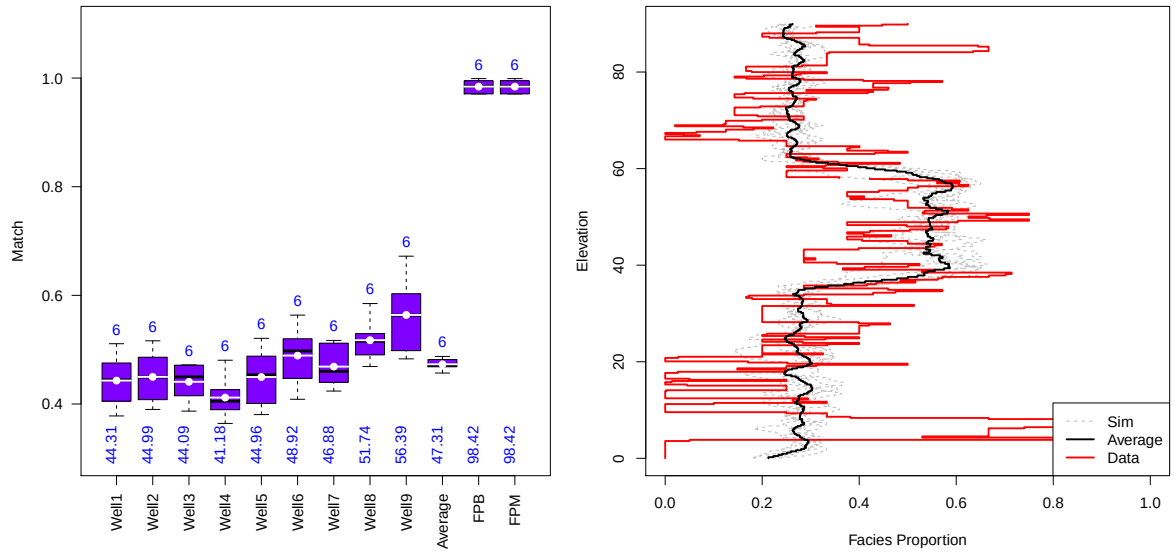


Figure 4.35: Results of the **sequential conditioning method** with the data and conditional simulations divided into **thirteen-facies wells** for the **Loranca case study** from 6 independent realizations. On the left, box plots of the facies matches for the 4 wells, their mean, the sand proportion balance and the sand proportion match. Right, the sand VPC of well data, of the conditional simulations and their average.

4.5 Conclusions

A sequential conditioning method based on particle filtering has been designed to generate conditional simulations of the Flumy process-based model. Unlike the original method (the dynamic conditioning), the sequential conditioning method can not only conditionally simulate under facies wells constraint, but also can integrate almost any kind of data (while it can be sequentially incorporated).

Both methods are tested in a synthetic and a real case. The first one uses 4 wells and the sand VPC as conditioning data, both extracted from the reference simulation. The sand VPC is directly measured from the block model. The second one uses 9 wells and the sand VPC as conditioning data. The sand VPC is computed from the wells. Both case studies are tested (i) by grouping the well facies in two categories (sand/clay) and (ii) by considering all available facies (no groups). Therefore, both methods are tested assuming to be known the simulation parameters. For the synthetic case, the model parameters are those used for the reference simulation, and for the real case, the model parameters are taken from the literature.

It is interesting to compare sections and $3D$ views. From a visual perspective, in the synthetic case, the conditional simulations obtained by the two methods, are comparable, and quite similar to the reference simulation (Figs. 4.6 and 4.7). With this visual criterion, the two methods give satisfactory results. In particular, no obvious distortion artifact can be observed for the dynamic conditioning as could be feared by modifying the processes. The similarity between the conditional simulations obtained by the two conditioning methods in the real case confirms these results.

In terms of quantitative performance, facies match at wells, in particular, the results are contrasted. Grouping the facies into two classes (sand-clay) results in a better match. Interestingly, the dynamic method gives results that are stable for the synthetic case and the real case, whether using 2 classes (85.15% and 87.69%) or 13 facies (57.77% and 59.21%). Compared to dynamic conditioning, sequential conditioning shows a better performance in the synthetic case, but a lower one in the real case. This probably comes from the fact that the number of particles was too low for this case, due to computational capacity restrictions.

The sequential method assimilates data while running Flumy layer by layer. At each step, the

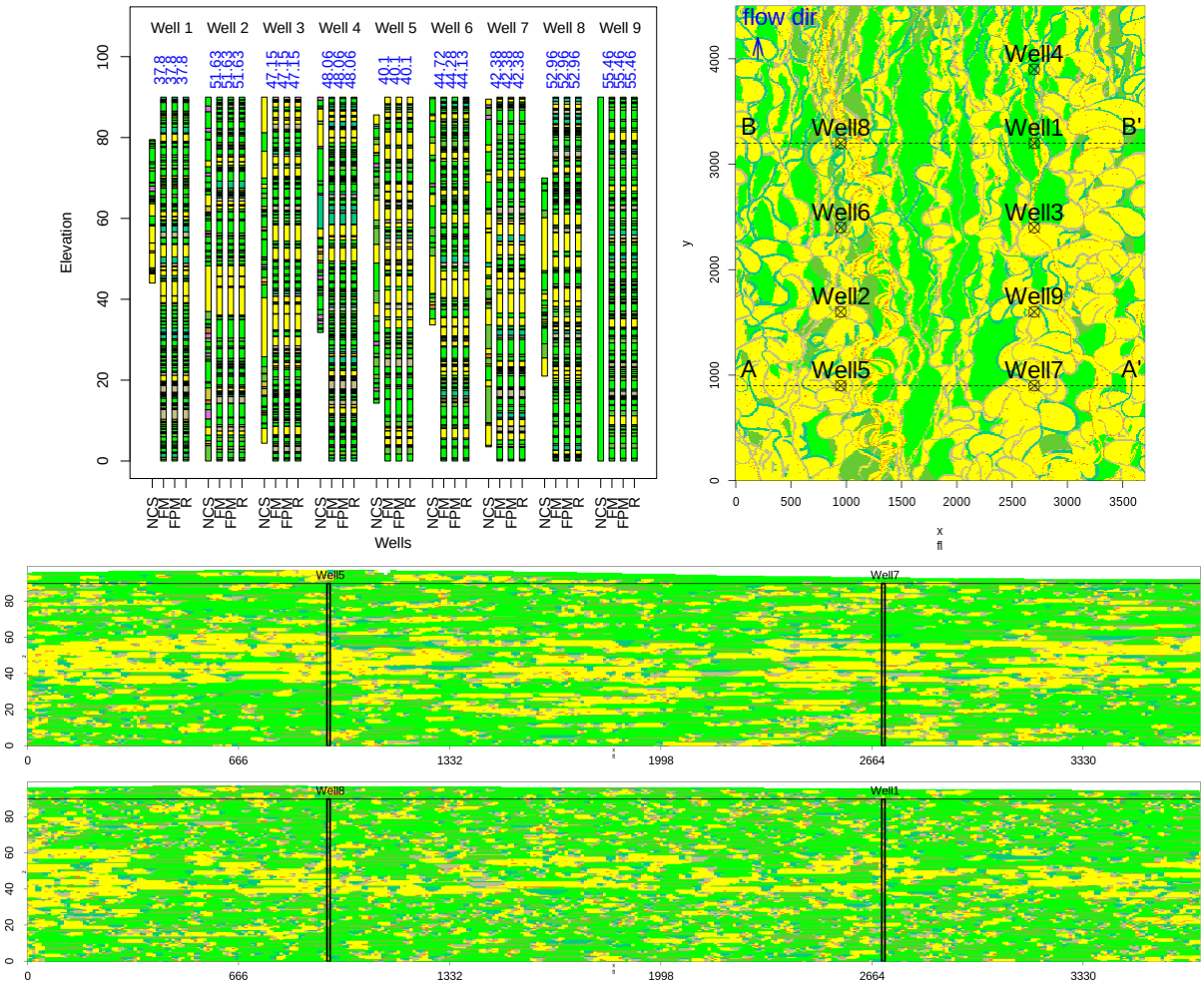


Figure 4.36: From one realization of the **sequential conditioning method** with **thirteen-facies wells** for the **Loranca case study**, three candidates are selected using the three criteria FM, FPM and R are shown on the top left. Top right, the horizontal slice (z=45m) within the second unit of the Flumy simulation for the candidate FM. In the middle and on the bottom, the cross-sections $A - A'$, and $B - B'$, respectively.

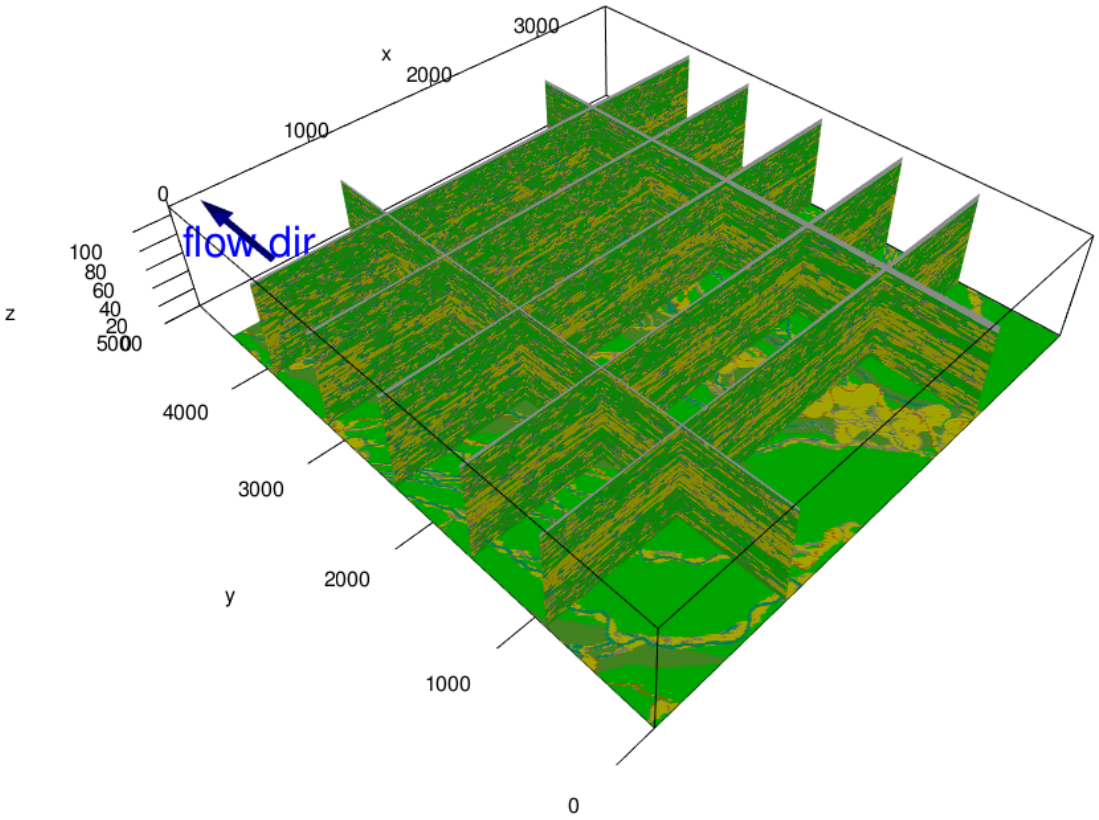


Figure 4.37: $3D$ view of the candidate FM using the **sequential conditioning method** with **thirteen-facies wells** for the **Loranca case study**.

so-called "particles" are generated, and the best ones are selected by resampling. Resampling weights can easily include different constraints. A sensitivity analysis performed in the two-facies synthetic case highlights the major role of the facies match within the current layer, but shows that accounting for the depositions located just above (which can be reworked) may also be beneficial. By contrast, adding a constraint on the sand VPC was not useful. The thickness of the layers was always fixed as one tenth of the maximal channel depth (Flumy parameter), but it would be interesting to make it varying and study its influence on the results.

Finally, the adoption of the sequential method to aggrading systems like Flumy seems relatively easy. In order to perform conditional simulations using this method, it is not needed to have a deep knowledge of the processes, since they are not modified. However, its major limitation concerns the time and memory required for the calculations. The performance of the method largely depends on the number of particles in each realization.

5

CONCLUSIONS AND PERSPECTIVES

Une nouvelle méthodologie basée sur le filtrage particulaire a été proposée et formalisée pour fournir des simulations conditionnelles. La méthode est générale, elle peut donc être utilisée dans une variété de situations. Dans cette thèse, la méthode a été appliquée à la simulation conditionnelle de modèles de réservoir. D'abord, elle est appliquée au schéma Booléen. Elle montre une reproduction parfaite des faciès, avec un temps de calcul court par rapport à la méthode traditionnelle itérative. Ensuite, la méthode est appliquée sur Flumy, un modèle pour des systèmes chenalisés méandriformes. Les résultats semblent prometteurs pour conditionner les simulations aux données.

In this work, we propose to use the sequential method called "particle filtering" to perform conditional simulations of spatial stochastic models (cf. chapter 2). This method consists in running a set of non-conditional simulations (the so called "particles") of the model, while assimilating the data sequentially and resampling the simulations that best match with the field constraints. The sequential method has been used to perform conditional simulations of two stochastic reservoir models, the Boolean model and the Flumy model.

In the case of the Boolean model, some independence properties of the model have been used to propose a new conditioning algorithm (cf. chapter 3) where the conditioning data points can be assimilated in an arbitrary order. This sequential conditioning appears to be much faster than the conditioning using the current MCMC algorithm. However, by contrast with the latter, the sequential conditioning may not be suited to other conditioning constraints such as connectivity constraints.

In the second case, the stochastic reservoir model is Flumy, a process-based model for meandering channelized systems (cf. chapter 4). This model mimics the evolution in time of the deposition of sediments transported by the system. A so called dynamic conditioning had previously been developed. This consists in modifying the processes (migration, aggradation, avulsions) in time in order to honor at best the data points. The development of such a dynamic conditioning is extremely difficult, as it requires a fine understanding of the processes in order to make ad-hoc adjustments, while reducing the risk of distortion of the bodies and of unbalance of the processes.

By contrast to the dynamic conditioning of Flumy model, the sequential conditioning does not require getting into the sedimentological processes. The great advantage is that it can be adapted easily to other aggrading systems, avoiding the hard task of developing an ad-hoc dynamic conditioning. It assimilates data sequentially by a continuous resampling of the best matching simulations among a set of non-conditional ones. The conditioning results can be improved by increasing the number of these non-conditional simulations, but the requirement in terms of computation time and memory are severe limitations of the method. Sequential conditioning and dynamic conditioning have been compared on both a synthetic case and a real

case study, using either all Flumy facies or grouping them into two classes (sand/clay). Both conditioning methods produce acceptable results, but they cannot ensure a 100% conditioning. The sequential conditioning matches better with data, particularly when using all the facies in a synthetic case. However, in a real case, the results are not good enough, but only because of the limitation of computational resources for the sequential method. It can also incorporate other constraints, such as a Vertical Proportion Curve of facies. Moreover, accounting for new conditioning constraints to the sequential method is quite easy.

The present work has focused on the conditioning of a stochastic model to data constraints. This does not include the inference of the model parameters, but only assumed that their values are compatible with the data. This is automatic for the synthetic examples of both Boolean and Flumy model, since the conditioning is using the parameters of the non-conditional model from where data originate. However, for the Loranca real case study, Flumy model parameters have been taken from earlier studies of the field (Weill et al., 2013).

In the future, Deep Learning could also be considered for conditioning stochastic reservoir models (Dupont et al., 2018). This approach consists in building a parameterized proxy of the non-conditional model. This does not provide better simulations than the original non-conditional simulations, but will certainly increase the speed of the conditioning procedure.

Acknowledgments: The author acknowledges the financial support of the Chile National Agency for Research and Development (ANID)/Scholarship Program/DOCTORADO BECAS CHILE/2018 - 72190309.

A

CONDITIONAL SIMULATION OF A BOOLEAN MODEL OF RESERVOIR USING PARTICLE FILTERING

A.1 Proof: Variogram of the Boolean model

First, consider a disc with a fixed radius r. Its geometric covariogram is

$$K_r(h) = 2r\left[\arccos\left(\frac{|h|}{2r}\right) - \frac{|h|}{2r}\sqrt{1-\frac{|h|^2}{4r^2}})\right]\mathbb{1}_{|h|\leq 2r}$$

Now, if the radius of the disc is randomly distributed according to the p.d.f. f, its geometric covariogram is expressed as

$$K_A(h) = \int_0^\infty K_r(h)f(r)\,dr$$

To derive a closed expression (Alfaro, 1979), the derivative is computed:

$$\frac{dK_A(h)}{d|h|} = -\int_{|h|/2}^\infty \sqrt{4r^2-|h|^2}f(r)\,dr$$

The change of variable $r = |h|/2\cosh t$

$$\frac{dK_A(h)}{d|h|} = -\frac{|h|^2}{2}\int_0^\infty \sinh^2 t\, f(\frac{|h|}{2}\cosh t)\,dt$$

If the radii are exponentially distributed with $f(r) = a\exp(-ar)$, then we have:

$$\frac{dK_A(h)}{d|h|} = -\frac{a|h|^2}{2}\int_0^\infty \sinh^2 t\,\exp\left(-\frac{a|h|}{2}\cosh t\right)\,dt.$$

From formula 9.6.23 of Abramovitz and Stegun (1970), we get

$$\frac{dK_A(h)}{d|h|} = -|h|K_1\left(\frac{a|h|}{2}\right)$$

It follows:

$$K_A(h) - K_A(0) = \int_0^{|h|} tK_0'\left(\frac{at}{2}\right)\,dt = \frac{4}{a^2}\int_0^{\frac{a|h|}{2}} tK_0'(t)\,dt$$

Which implies by integration by parts

$$K_A(h) - K_A(0) = \frac{4}{a^2}\left[\frac{a|h|}{2}K_0\left(\frac{a|h|}{2}\right) - \int_0^{\frac{a|h|}{2}} K_0(t)\,dt\right]$$

This last integral has an explicit expression:

$$\int_0^{\frac{a|h|}{2}} K_0(t)\,dt = \frac{a|h|}{2}K_0\left(\frac{a|h|}{2}\right) + \frac{a|h|\pi}{4}\left[L_0\left(\frac{a|h|}{2}\right)K_1\left(\frac{a|h|}{2}\right) + L_1\left(\frac{a|h|}{2}\right)K_0\left(\frac{a|h|}{2}\right)\right]$$

where $L_\nu(t)$ is the modified Struve function:

$$L_\nu(t) = \frac{t}{2}^{\nu+1}\sum_{k=0}^{\infty}\frac{\frac{t}{2}^{2k}}{\Gamma(k+\frac{3}{2})\Gamma(k+\nu+\frac{3}{2})}$$

Finally, the following has been obtained:

$$K_A(0) - K_A(h) = \frac{\pi|h|}{a}\left[K_0\left(\frac{a|h|}{2}\right)L_1\left(\frac{a|h|}{2}\right) + K_1\left(\frac{a|h|}{2}\right)L_0\left(\frac{a|h|}{2}\right)\right].$$

A.2 Sequential conditional simulation results with 400, 800, 1600, and 3200 particles

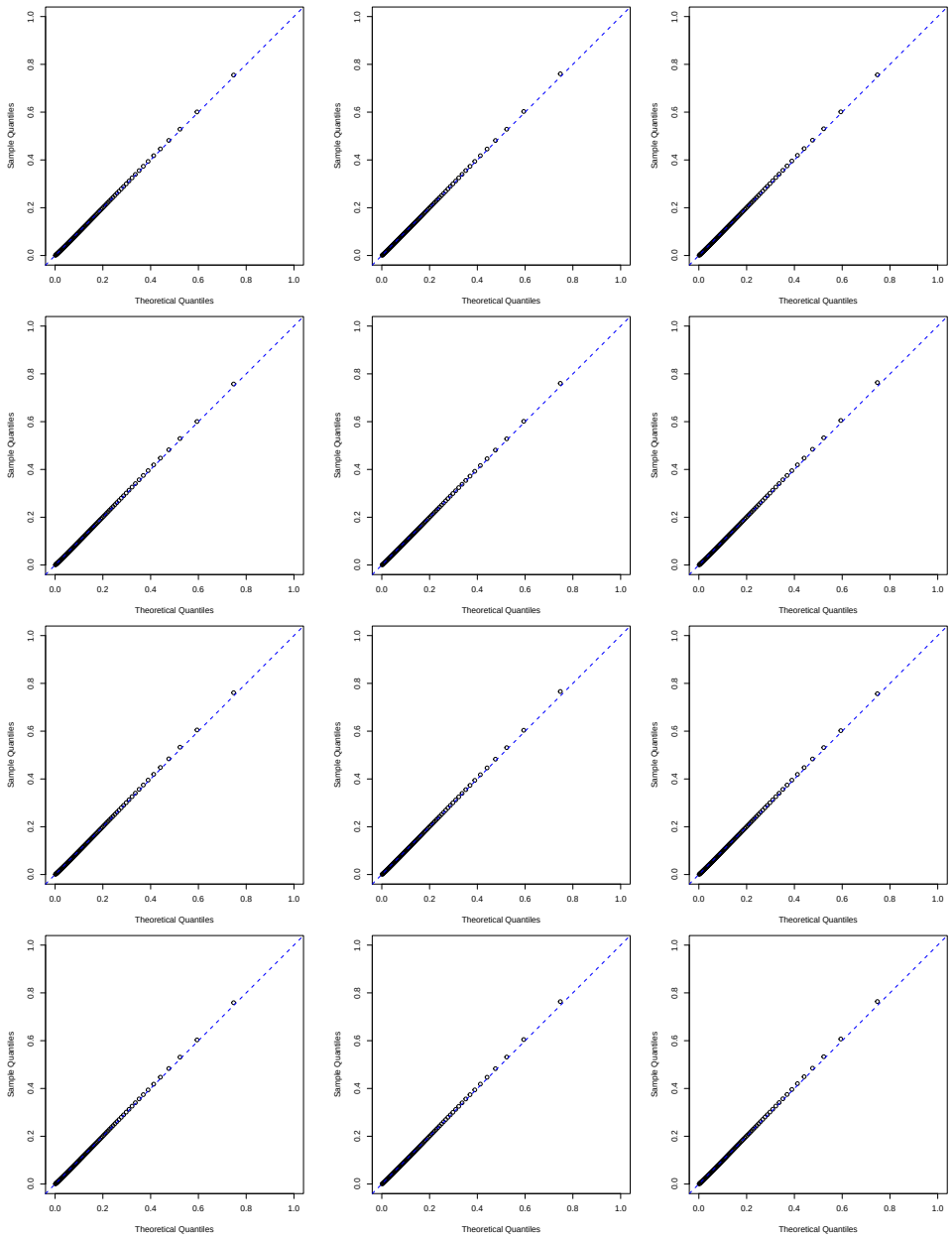


Figure A.1: QQ-plot of the disc radii of the result of 500 simulations of the conditional Boolean model with the sequential algorithm. On the left, purely random point fixed with random point values, in the middle, structured point fixed with random point values, and on the right, random point positions, and random point values. From top to bottom, 400, 800, 1600, and 3200 particles.

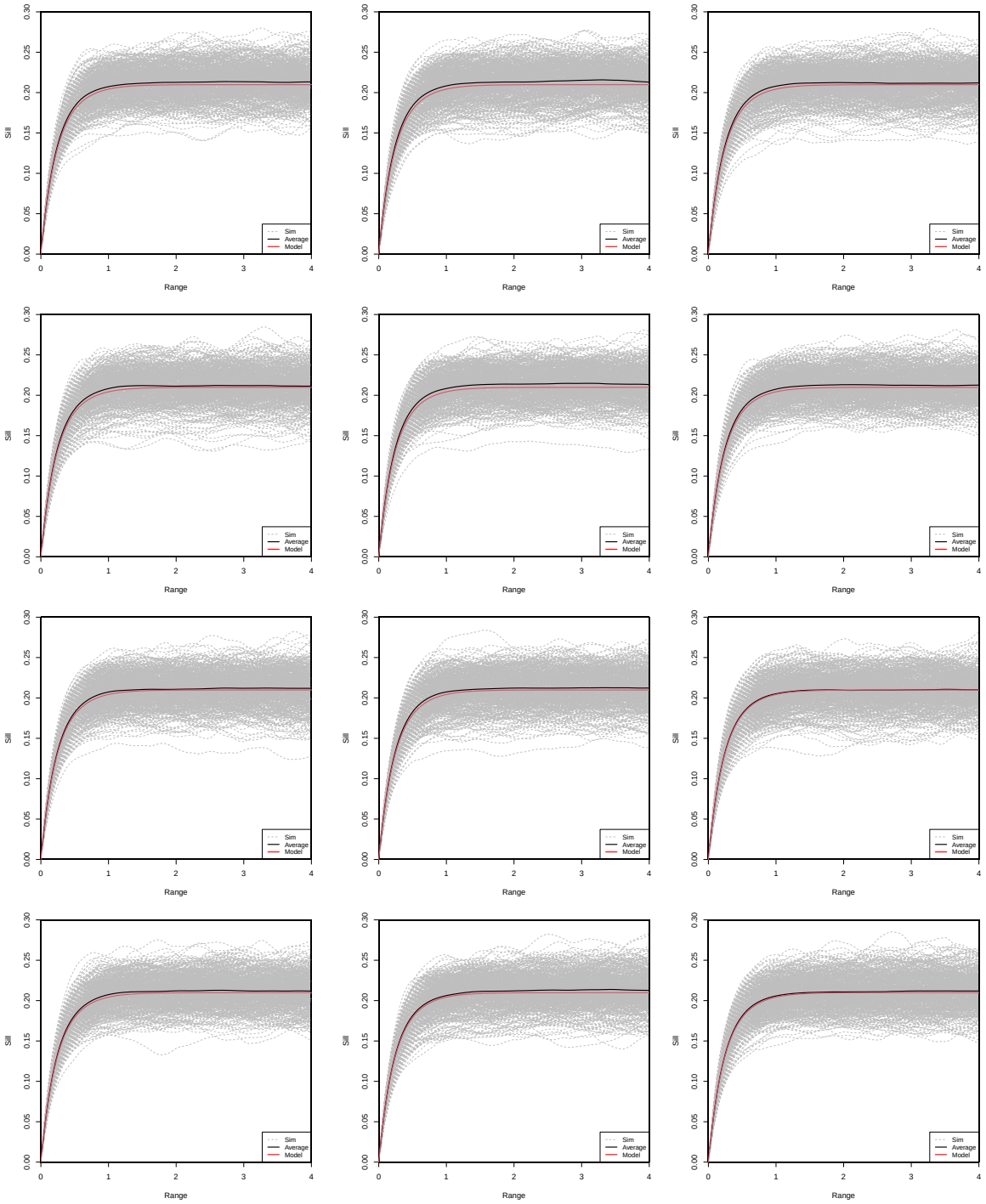


Figure A.2: Variogram of each simulation with its average and model of the disc radii of the result of 500 simulations of the conditional Boolean model with the sequential algorithm. On the left, purely random point fixed with random point values, in the middle, structured point fixed with random point values, and on the right, random point positions, and random point values. From top to bottom, 400, 800, 1600, and 3200 particles.

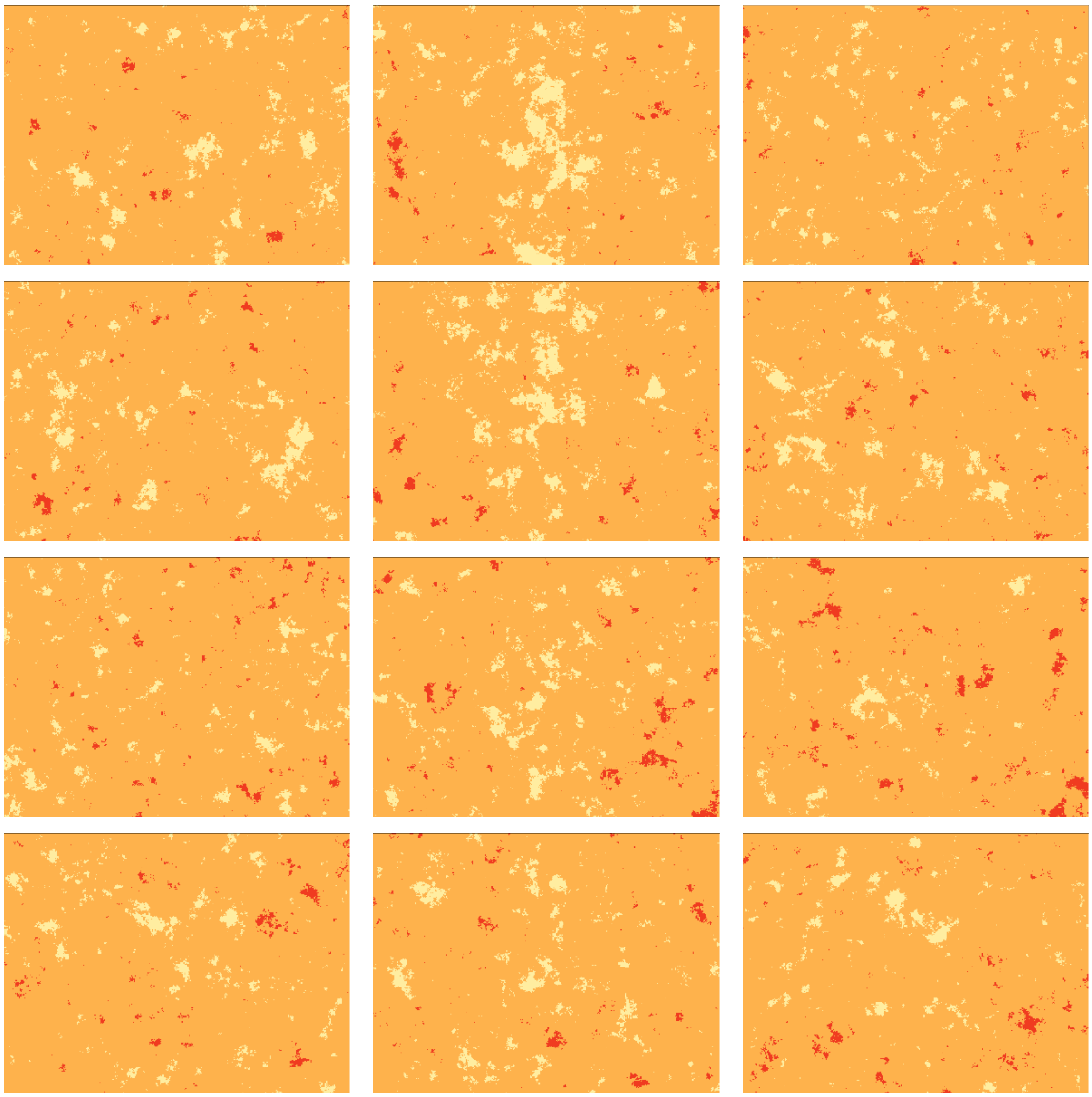


Figure A.3: Probability map with a threshold of two standard deviations of the result of 500 simulations of the conditional Boolean model with the sequential algorithm. On the left, purely random point fixed with random point values, in the middle, structured point fixed with random point values, and on the right, random point positions, and random point values. From top to bottom, 400, 800, 1600, and 3200 particles.

B

CONDITIONAL SIMULATION OF A PROCESS-BASED RESERVOIR MODEL USING PARTICLE FILTERING

B.1 Sensitivity analysis of the sequential conditioning method

This section shows different combinations of the parameters α, β and γ varying μ and p for both criteria: Min and Average.

The box plots show that:

- The more particles are used, the better the matches in terms of mean well facies match, regardless of the criteria chosen.

- Adding the influence of the unconsolidated zone (μ with small values or β important regarding α and γ) and of the sand proportion ($\gamma \neq 0$) increases the amount of *different common ancestors*[1] (or different ancestors). Different ancestors mean that: when we select a particle according to different criteria, we have more chance to take candidates with very different evolution histories.

- Adding the influence of γ does not change much the values of the sand proportion balance regarding the cases with $\gamma = 0$. All the mean values of the sand proportion balance are around 100%. However, the influence of γ decreases the mean well facies match.

- When more different ancestors are involved, the facies proportion balance improves with more particles, since different reservoir histories can be considerated for the selecting criteria.

- The Average criterion gives lower results in terms of mean well facies match than the Min criterion for the same parameter combinations.

- The Min criterion has more chance to stop before finishing the simulation (cf. box plots with $p = 10$, $\gamma = 0$ and Min criterion, less than 25 realizations finished).

[1] We define the *evolution history* of a candidate, after finishing the particle filtering, as the path that this candidate had to endure to arrive to the end. When we look at the start of this history, we find its *ancestor*. The same ancestor may be shared by different candidates. In this case, they share part of their history.

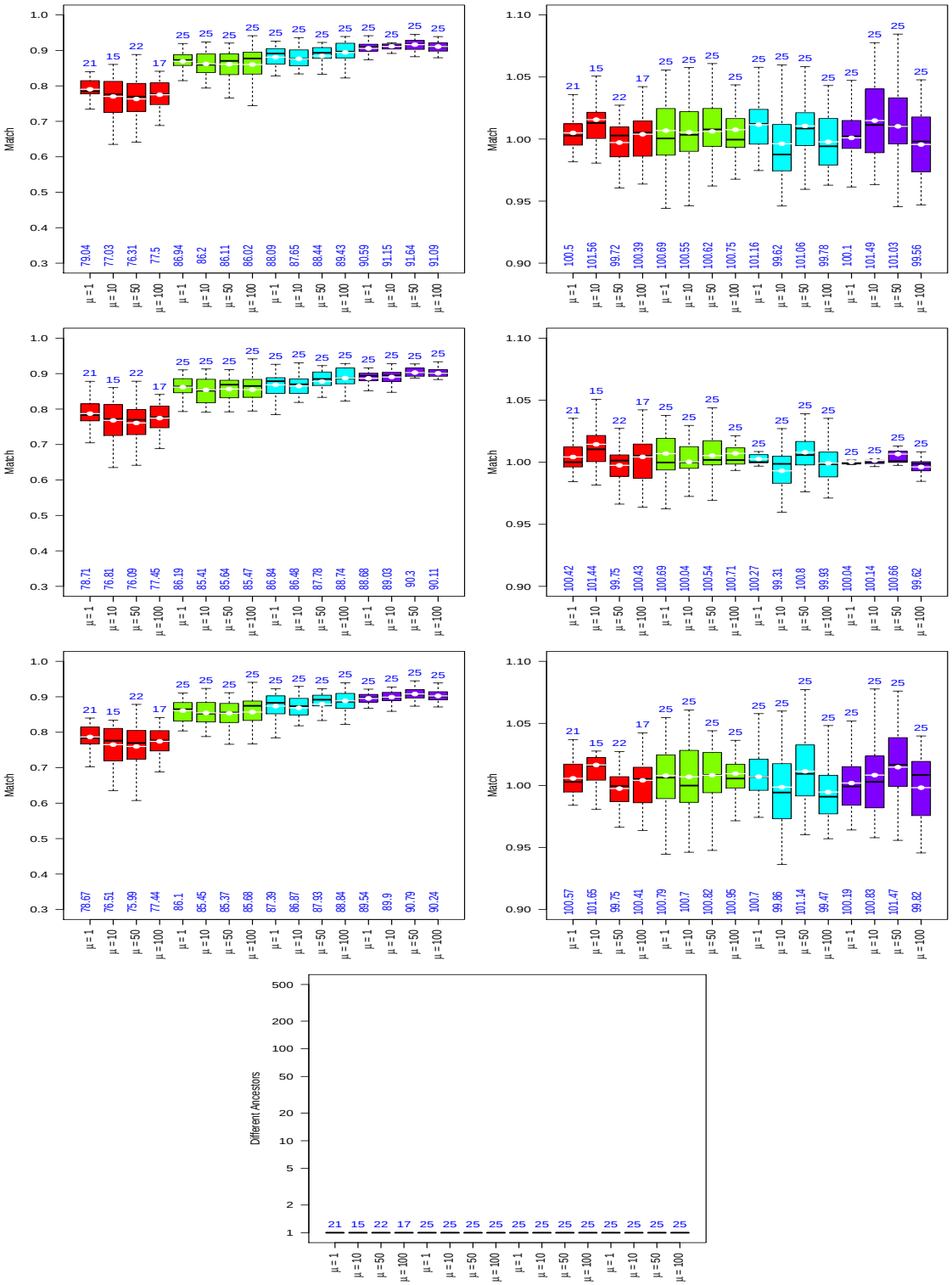


Figure B.1: (optimal parameters) Box plots of independent realizations of the sequential method using $\alpha = 0.8$, $\beta = 0.2$ and $\gamma = 0$ with the Min criterion. On the left, the mean well facies match and on the right, the mean facies proportion balance. For each realization, three candidates can be selected using the criteria (4.19), (4.20) and (4.21). In the first row, using FM, in the second row, using FPM, in the third row, R. Bottom, the number of different ancestors. Number of particles used: 10 (red), 50 (green), 100 (blue), and 250 (purple) in abscissa. For each number, $\mu = 1, 10, 50, 100$ (see Sect. 4.3.4.3).

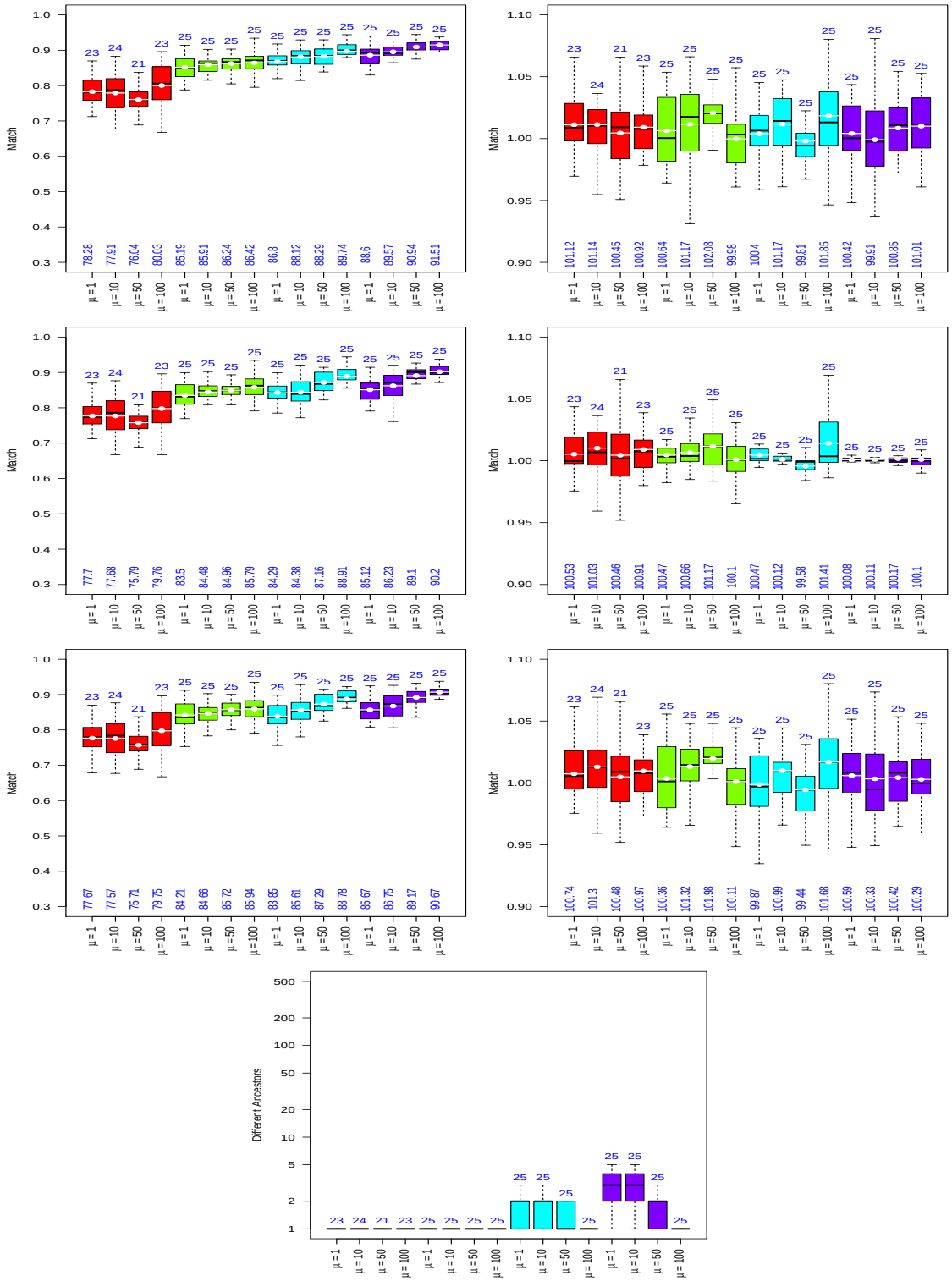


Figure B.2: Box plots of independent realizations of the sequential method using $\alpha = 0.5$, $\beta = 0.5$ and $\gamma = 0$ with the Min criterion (details in Fig. B.1).

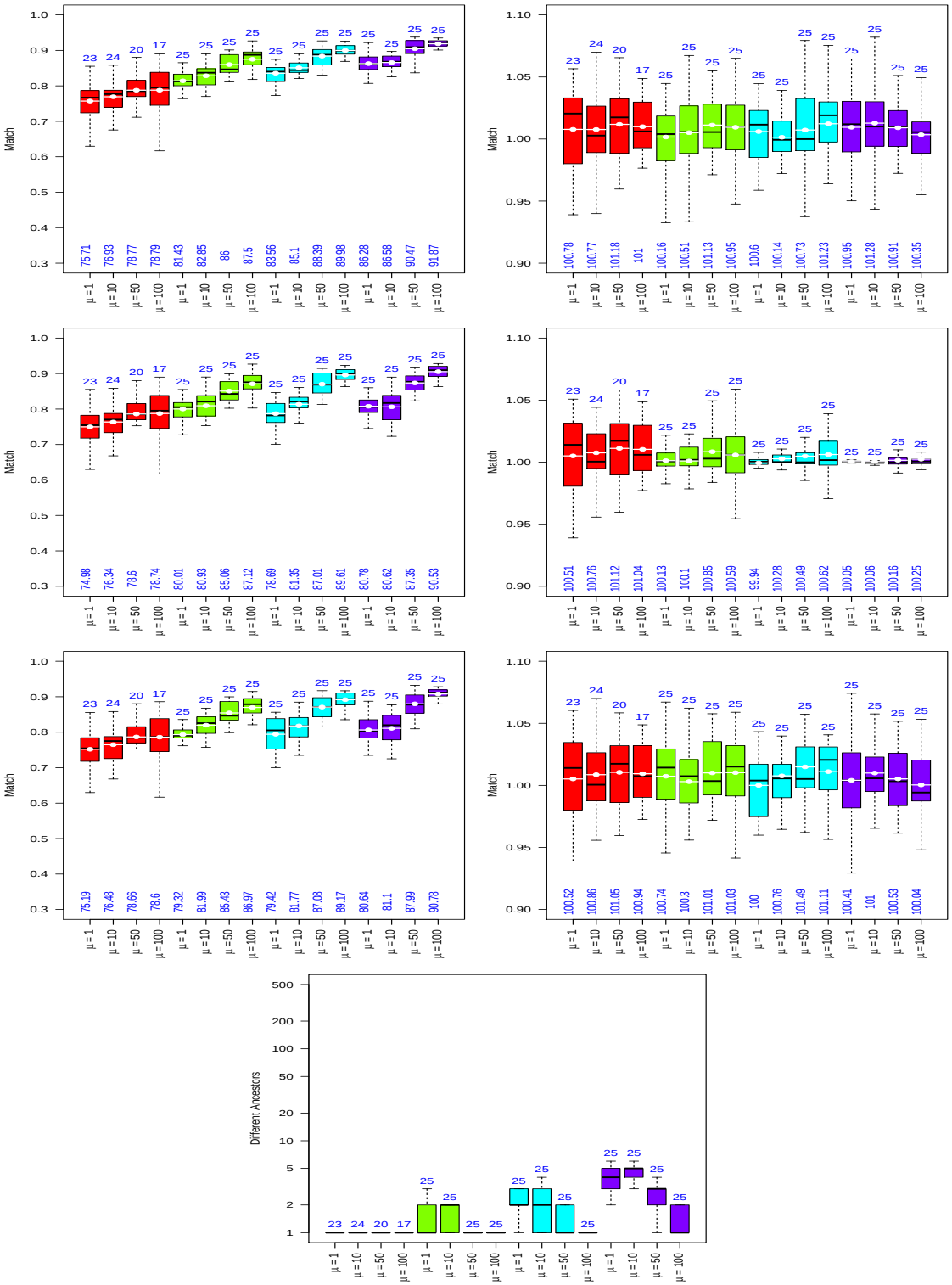


Figure B.3: Box plots of independent realizations of the sequential method using $\alpha = 0$, $\beta = 1$ and $\gamma = 0$ with the Min criterion (details in Fig. B.1).

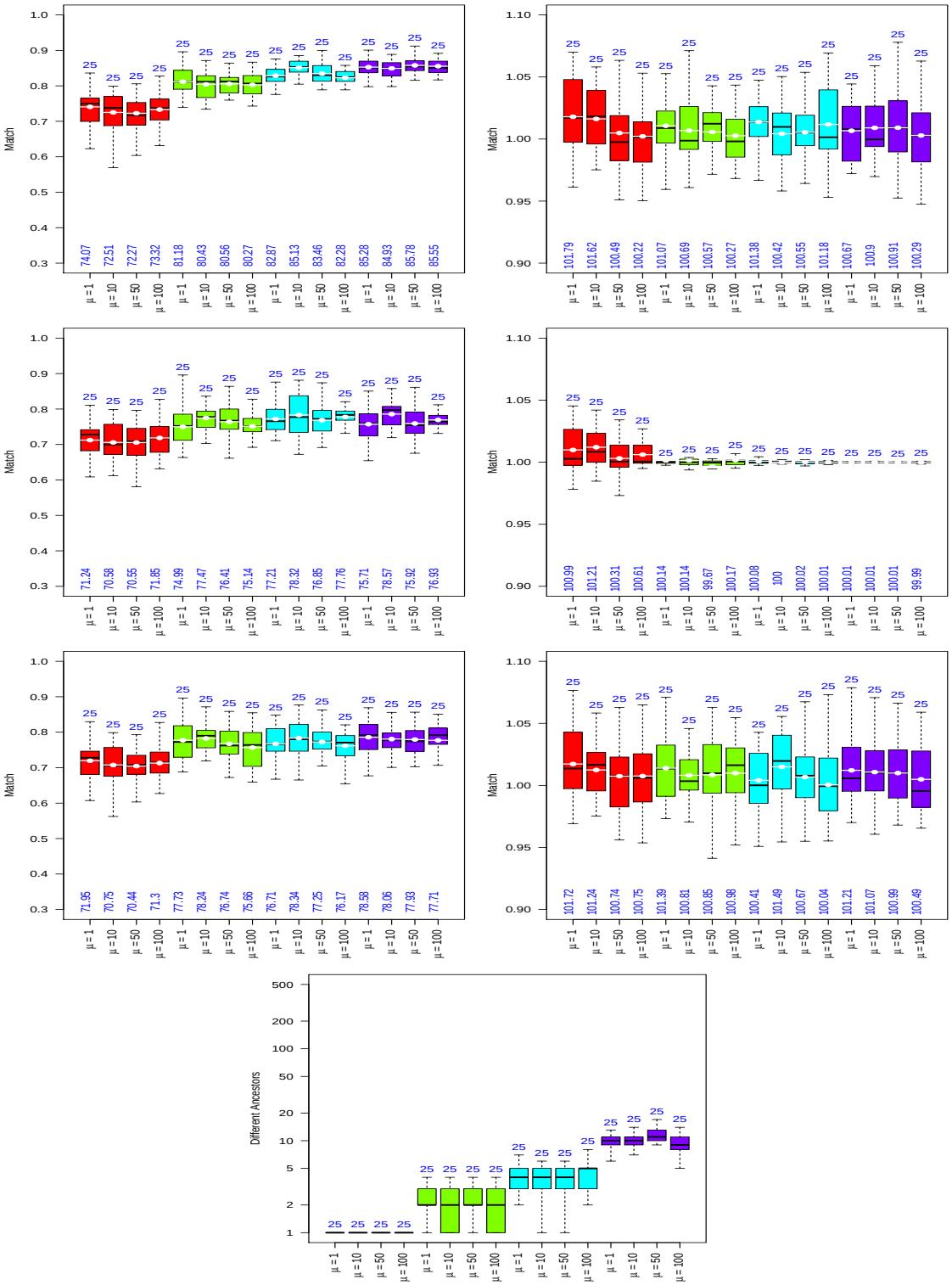


Figure B.4: Box plots of independent realizations of the sequential method using $\alpha = 1$, $\beta = 0$ and $\gamma = 0$ with the Average criterion (details in Fig. B.1).

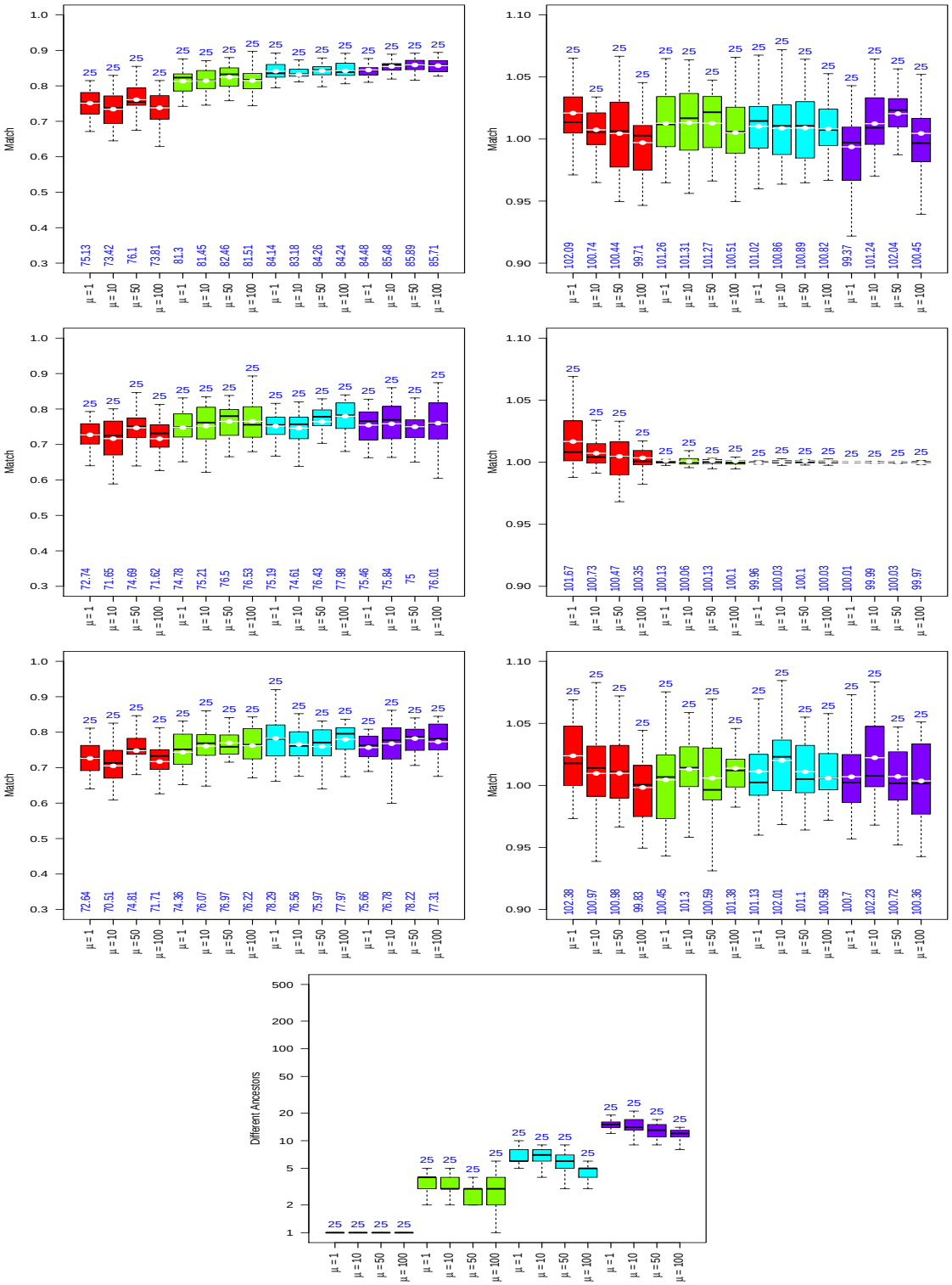


Figure B.5: Box plots of independent realizations of the sequential method using $\alpha = 0.5$, $\beta = 0.5$ and $\gamma = 0$ with the Average criterion (details in Fig. B.1).

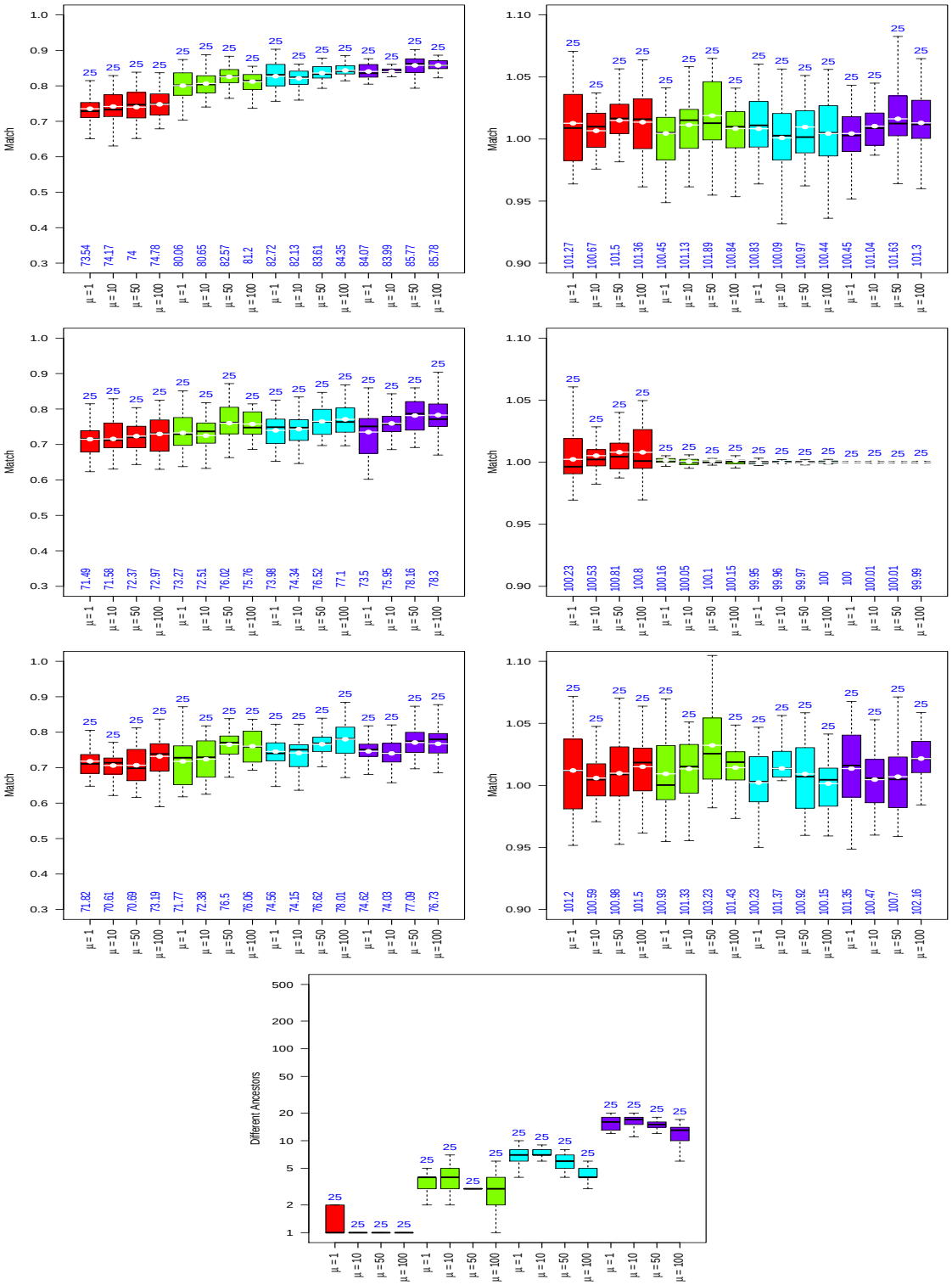


Figure B.6: Box plots of independent realizations of the sequential method using $\alpha = 0$, $\beta = 1$ and $\gamma = 0$ with the Average criterion (details in Fig. B.1).

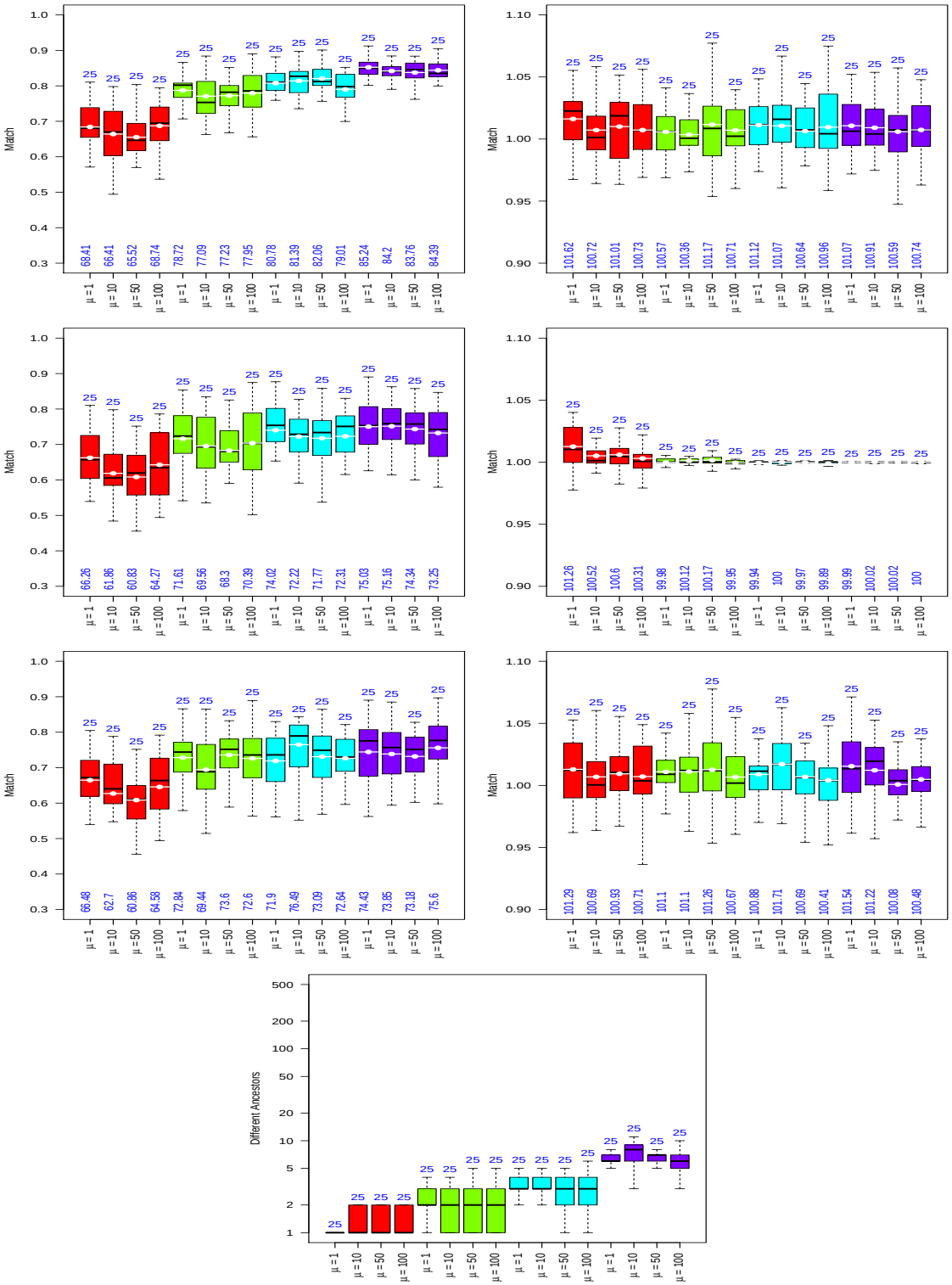


Figure B.7: Box plots of independent realizations of the sequential method using $\alpha = 0.5$, $\beta = 0$ and $\gamma = 0.5$ with the Min criterion (details in Fig. B.1).

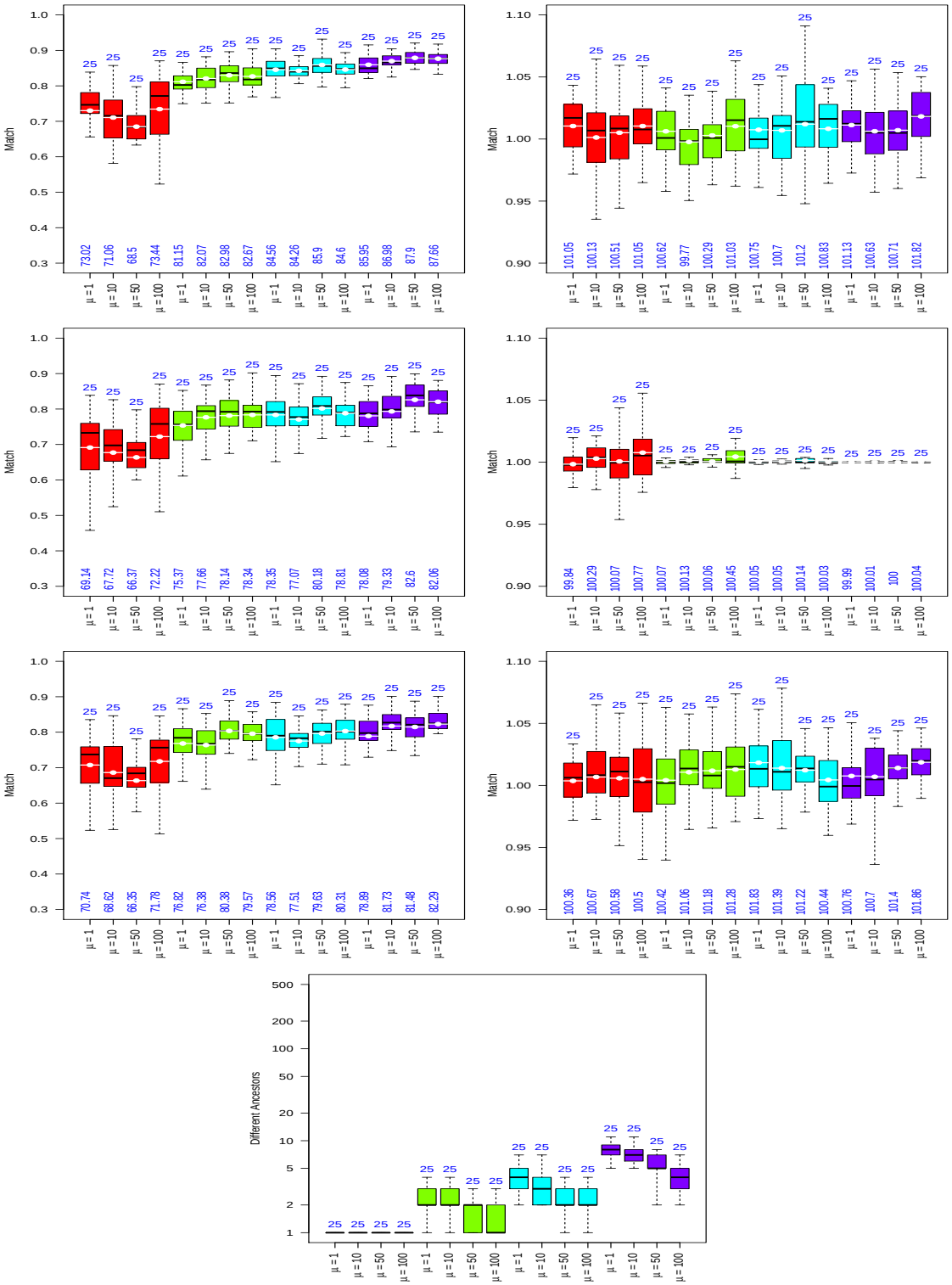


Figure B.8: Box plots of independent realizations of the sequential method using $\alpha = 0.\overline{3}$, $\beta = 0.\overline{3}$ and $\gamma = 0.\overline{3}$ with the Min criterion (details in Fig. B.1).

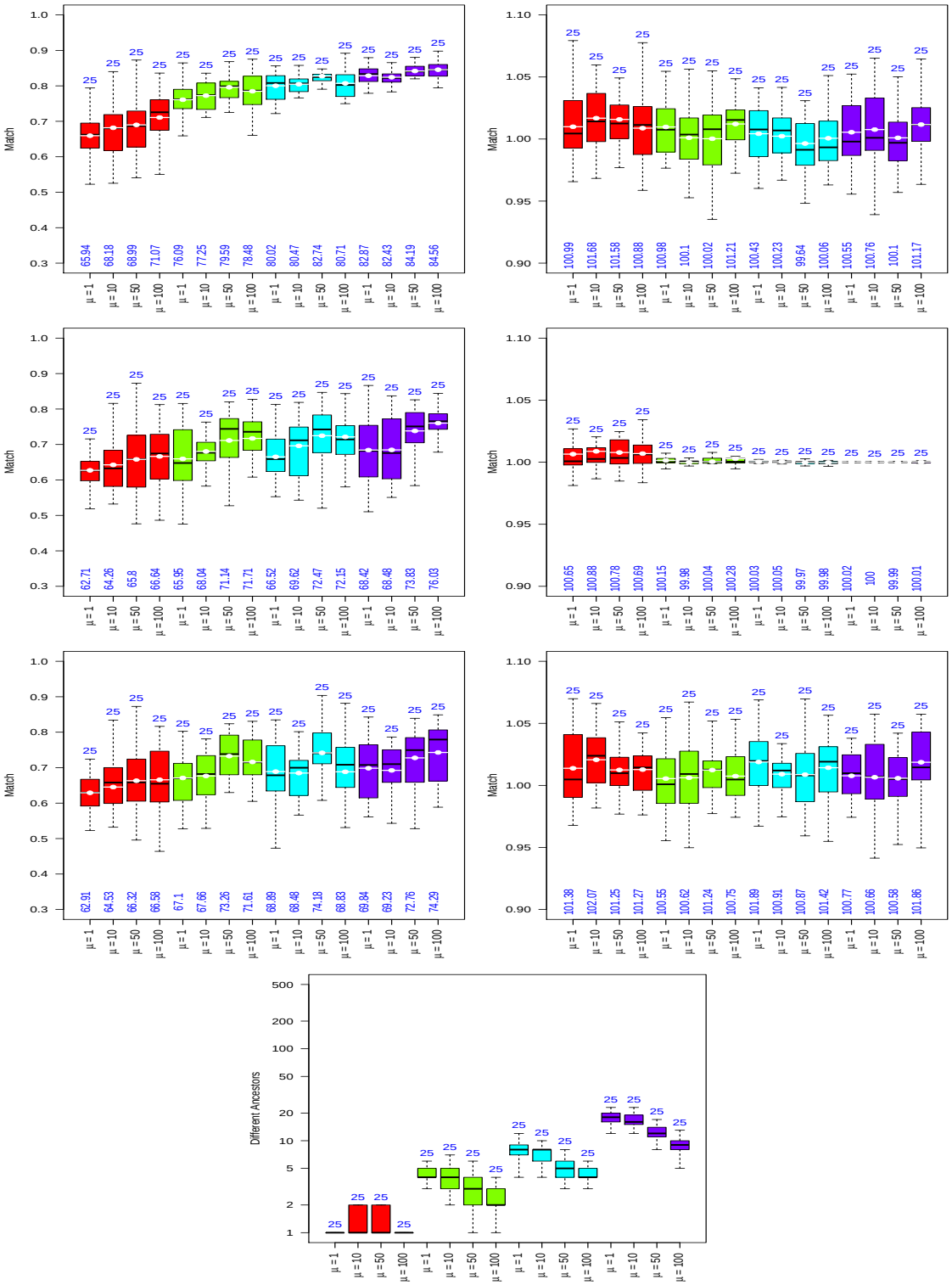


Figure B.9: Box plots of independent realizations of the sequential method using $\alpha = 0$, $\beta = 0.5$ and $\gamma = 0.5$ with the Min criterion (details in Fig. B.1).

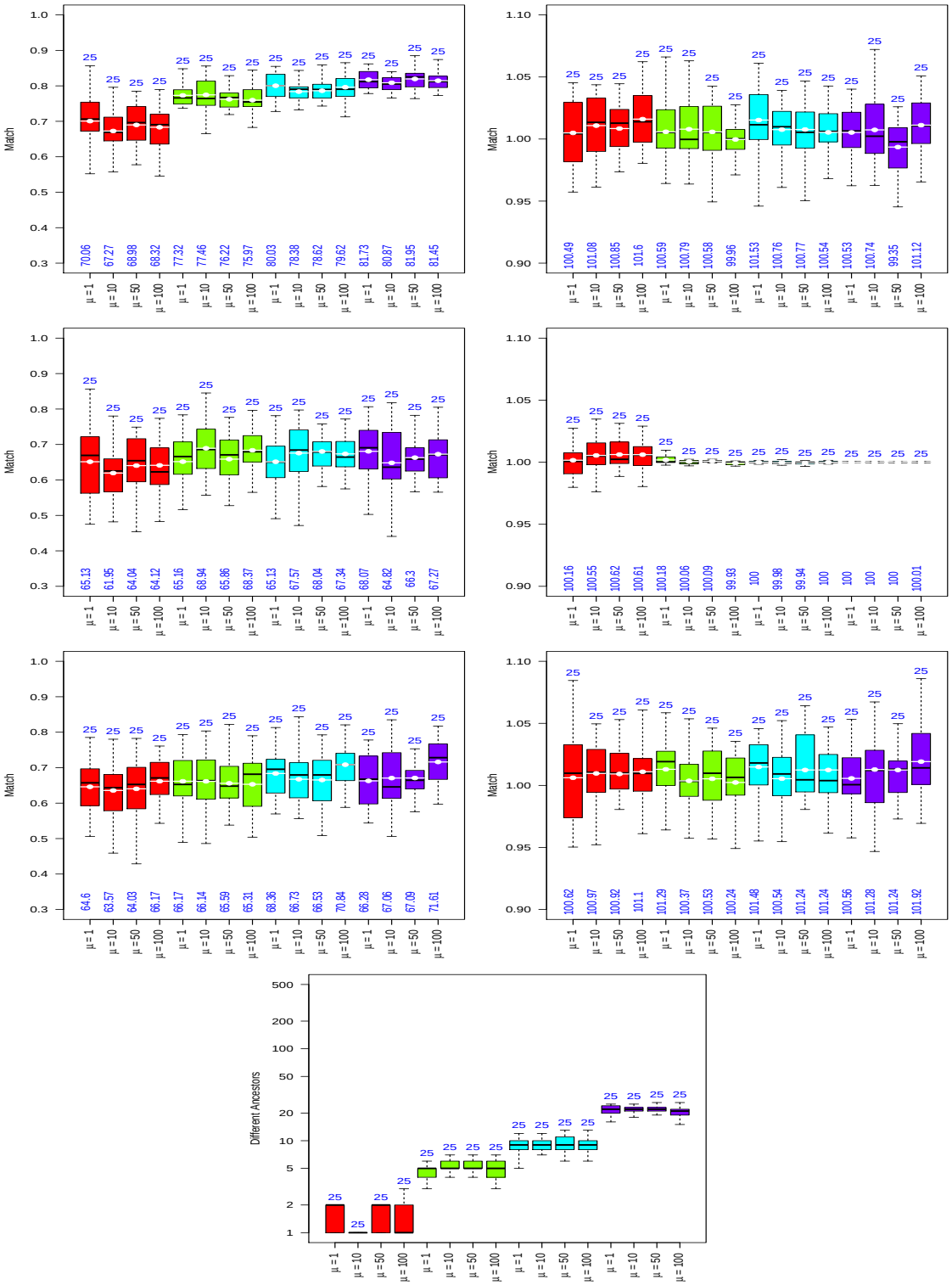


Figure B.10: Box plots of independent realizations of the sequential method using $\alpha = 0.5$, $\beta = 0$ and $\gamma = 0.5$ with the Average criterion (details in Fig. B.1).

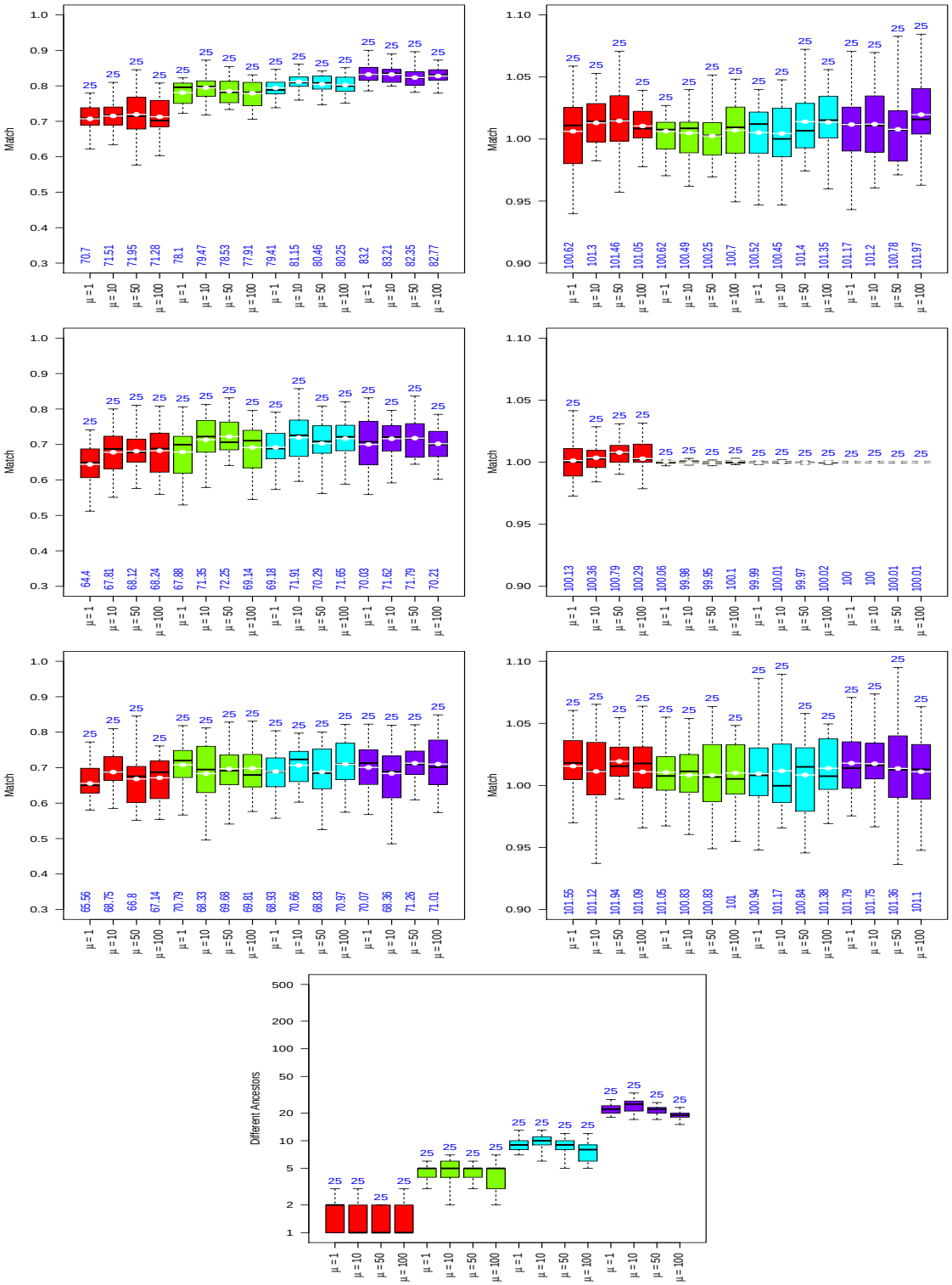


Figure B.11: Box plots of independent realizations of the sequential method using $\alpha = 0.\overline{3}$, $\beta = 0.\overline{3}$ and $\gamma = 0.\overline{3}$ with the Average criterion (details in Fig. B.1).

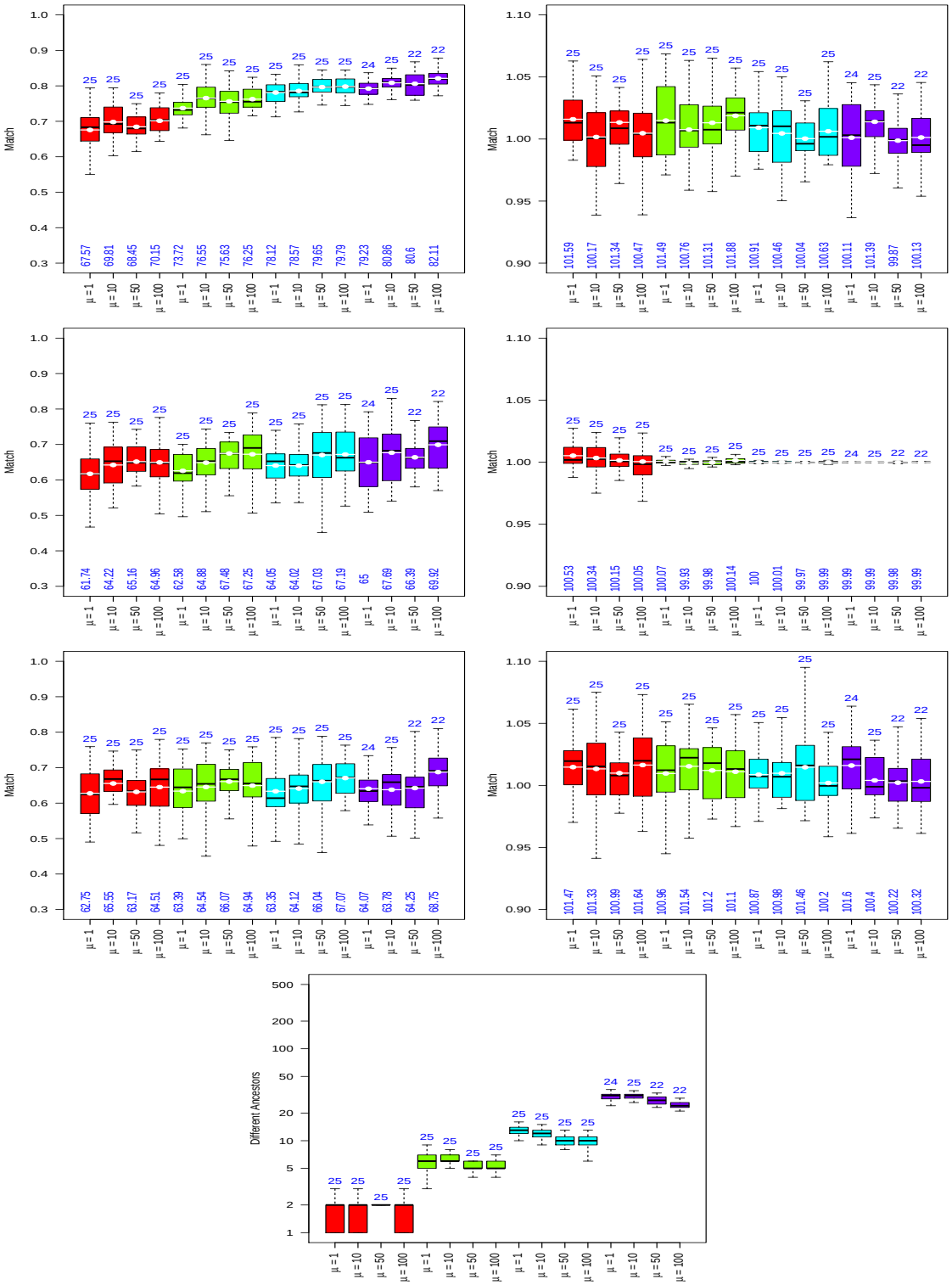


Figure B.12: Box plots of independent realizations of the sequential method using $\alpha = 0$, $\beta = 0.5$ and $\gamma = 0.5$ with the Average criterion (details in Fig. B.1).

B.2 Flumy Parameters: Loranca basin

Table B.1: Main Flumy parameters for the different Sedimentary Units (S.U.) for Loranca basin. More details in Flumy-Userguide (2022).

Parameter	S.U.1	S.U.2	S.U.3	Description
CHNL_MAX_DEPTH	2.39	2.98	2.29	Channel maximum bankfull depth [m]
CHNL_WIDTH	23.9	29.8	22.9	Channel bankfull width [m]
CHNL_WAVELENGTH	300	375	288	Mean wavelength before first neck cutoff [m]
EROD_COEF	$3.58\,10^{-8}$	$4.00\,10^{-8}$	$3.51\,10^{-8}$	Erodibility coefficient [m/s]
AG_OB_POISSON	19	60	18	Poisson parameter for the overbank period [number of iterations]
AG_OB_NORM_MEAN	$1.33\,10^{-1}$	$1.14\,10^{-1}$	$1.36\,10^{-1}$	Overbank thickness distribution mean [m]
AG_OB_NORM_STDEV	$3.32\,10^{-2}$	$3.46\,10^{-2}$	$3.59\,10^{-2}$	Overbank thickness distribution standard deviation [m]
AG_EXP_DEC_THICK	722	852	690	Overbank thickness exponential decrease [m]
AG_LV_WIDTH	6	6	6	Levee width (multiple of the channel width)
AV_LOC_FREQ	None	None	None	Local Avulsion frequency
AV_LV_OB	TRUE	TRUE	TRUE	Levee breaches during aggradation
AV_REG_POISSON	1000	1000	1000	Poisson parameter for the regional avulsion period [number of iterations]
AV_LOC_PROB1	0.5	0.3	0.5	Probability for transition from Crevasse Splay I to Crevasse Splay II
AV_LOC_PROB2	0.1	0.1	0.1	Probability for adding a new Crevasse Splay Channel

B.3 Glossary

p Number of particles running in parallel at each step of the sequential method.

j Index of a particle varying from 1 to p.

L Number of wells.

ℓ Index of a well varying from 1 to L.

k Number of steps for the sequential simulation.

H_{max} Maximum depth of the meandering channel.

i Index of the current step of the sequential method varying from 1 to k. It can also represent the consolidated layer $[z_i, z_{i+1}]$ or the unconsolidated zone $[z_{i+1}, z_{i+1} + H_{max}]$.

$I = [zb, zt]$ A generic vertical interval, $zb < zt$.

$DZ = [z_{min}, z_{max}]$ Interval given by the domain vertical limits.

z_i Bottom limit of the consolidated layer for the current step i.

z_{i+1} Upper limit of the consolidated layer for the current step i. At the same time, it is the bottom limit of the unconsolidated zone at the same step.

$z_{i+1} + H_{max}$ Upper limit of the unconsolidated zone used at the current step i.

r Ratio between the migration and the aggradation forecast rates.

z_{ul} Upper limit elevation that must be exceeded everywhere by the topography in order to stop simulation.

$F_\ell(z)$ Facies data at elevation z in well ℓ.

$F_\ell^{(j)}(z)$ Facies deposited by simulation j at elevation z at the location of well ℓ.

$\psi_\ell^{(j)}(z)$ A weighting function of z to compute facies or grain size matches at well ℓ for simulation j.

$FM_{I,\ell}^{(j)}$ Well facies match between data and simulation j for well ℓ in a generic vertical interval I of the consolidated layer.

$FM_{i,\ell}^{(j)}$ Well facies match between data and simulation j for well ℓ at the current step i in its consolidated layer.

$FMU_{i,\ell}^{(j)}$ Well facies match between data and simulation j for well ℓ at the current step i in its unconsolidated zone.

$FM_{DZ,\ell}^{(j)}$ Well facies match between data and simulation j for well ℓ in the reservoir domain.

$FM_{I,average}^{(j)}$ Mean facies match of the L wells weighted by the well lengths in simulation j for a generic vertical interval I.

$FM_{i,average}^{(j)}$ Mean facies match of the L wells weighted by the well lengths in simulation j at the current step i in its consolidated layer.

$FMU_{i,average}^{(j)}$ Mean facies match of the L wells weighted by the well lengths in simulation j at the current step i in its unconsolidated zone.

$FM^{(j)}_{DZ,average}$ Mean facies match of the L wells weighted by the well lengths in simulation j in the reservoir domain.

$FM^{(j)}_{I,min}$ Minimum facies match over the L wells in simulation j in a generic vertical interval I of the consolidated layer.

$FM^{(j)}_{i,min}$ Minimum facies match over the L wells in simulation j at the current step i in its consolidated layer.

$FMU^{(j)}_{i,min}$ Minimum facies match over the L wells in simulation j at the current step i in its unconsolidated zone.

$G_\ell(z)$ Grain size class data at elevation z in well ℓ.

$G^{(j)}_\ell(z)$ Grain size class deposited by simulation j at elevation z at the location of well ℓ.

$GM^{(j)}_\ell$ Grain size class match between data and simulation j for well ℓ in the reservoir domain.

$GM^{(j)}_{average}$ Mean grain size class match of the L wells weighted by the well lengths in simulation j in the reservoir domain.

$FP(z)$ A facies proportion at elevation z (for instance, sand proportion deduced from a seismic block).

FP_I A facies proportion data in a generic vertical interval I.

FP_i A facies proportion data in the layer of the current step i.

FP_{DZ} A facies proportion data in the reservoir domain.

N/G Sand proportion data in the reservoir domain.

$FP^{(j)}(z)$ A facies proportion deposited by simulation j at elevation z.

$FP^{(j)}_I$ A facies proportion deposited by simulation j in a generic vertical interval I.

$FP^{(j)}_i$ A facies proportion deposited by simulation j for the current step i in its layer.

$FP^{(j)}_{DZ}$ A facies proportion deposited by simulation j in the reservoir domain.

$FPM^{(j)}_I$ A facies proportion match between data and simulation j in a generic vertical interval I.

$FPM^{(j)}_i$ A facies proportion match between data and simulation j at the current step i in its layer.

$FPM^{(j)}_{DZ}$ A facies proportion match between data and simulation j in the reservoir domain.

$FPB^{(j)}$ A facies proportion balance between data and simulation j in the reservoir domain.

FM Simulation that has the highest weighted mean facies match from a simulation set .

FPM Simulation that honors at best the facies proportion from a simulation set.

R A simulation chosen randomly from a simulation set.

B.4 Box plots of non-conditional simulations

Using a series of non-conditional simulations produce results with lower quality in terms of well facies matches than the iterative and sequential conditioning methods. On the other hand, the facies proportion balance and match is good since the simulations use the same parameters as the reference simulation.

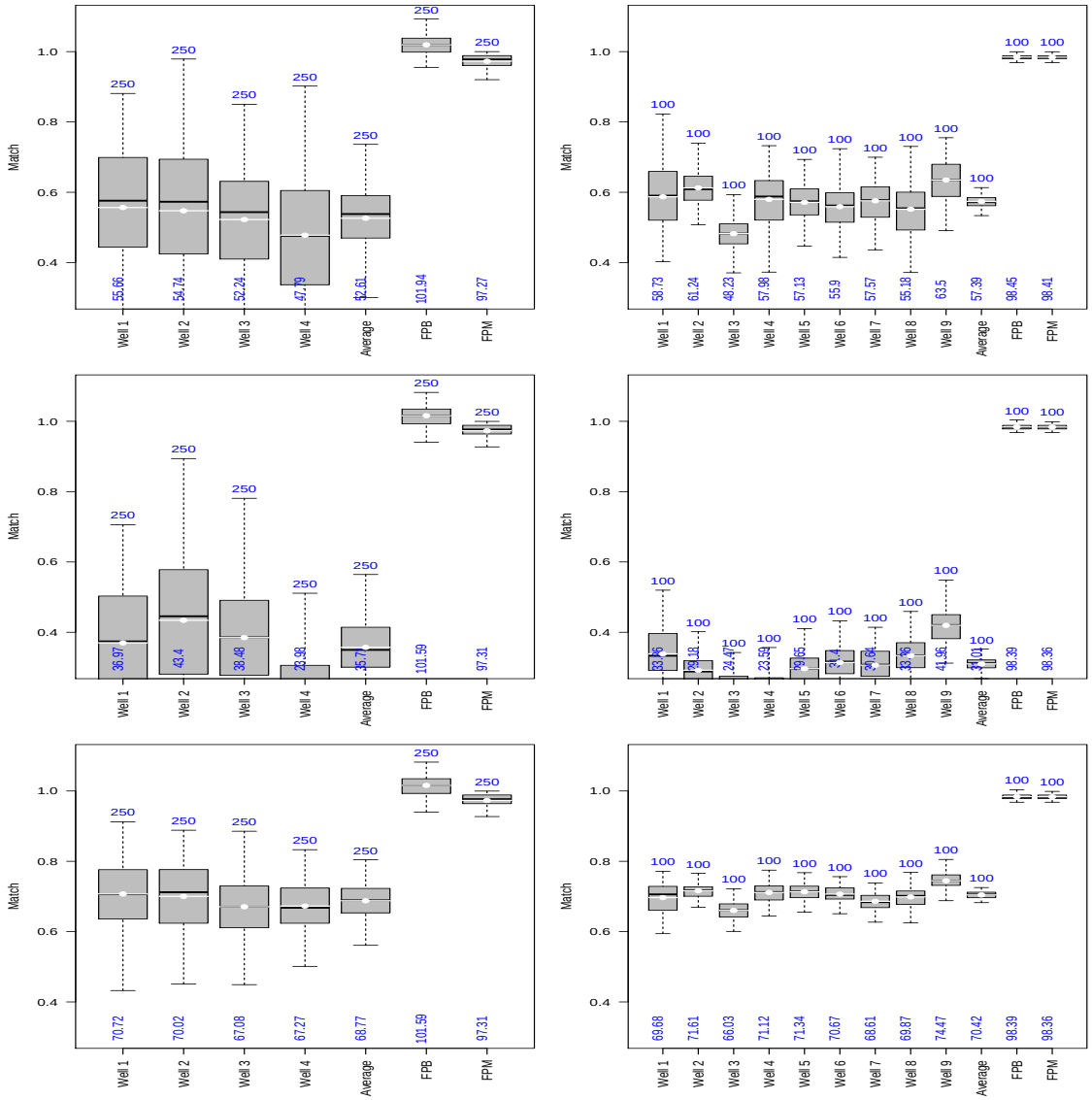


Figure B.13: Box plots of mean facies match and mean grain size class match of independent realizations of non-conditional simulations. Left, the **synthetic case** and right, **Loranca case study**. Top, mean facies match in the two-facies case, middle, mean facies match in the thirteen-facies case, and bottom, grain size class match in the thirteen-facies case.

BIBLIOGRAPHY

Abramovitz, M. and Stegun, I. (1970). *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*. Dover Publications, Inc., New York.

Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2017). Importance sampling: Intrinsic dimension and computational cost. *Statistical Science*, pages 405–431.

Alfaro, M. (1979). *Etude de la robustesse des simulations de fonctions aléatoires*. PhD thesis, Doctoral thesis, ENS des Mines de Paris.

Allard, D., Froidevaux, R., and Biver, P. (2006). Conditional simulation of multi-type non-stationary Markov object models respecting specified proportions. *Mathematical geology*, 38(8):959–986.

Armstrong, M. (1984). Problems with universal kriging. *Journal of the International Association for Mathematical Geology*, 16(1):101–108.

Armstrong *et al.*, M. (2011). *Plurigaussian Simulations in Geosciences*. Springer, Berlin.

Barbier, E. (2002). Geothermal energy technology and current status: an overview. *Renewable and sustainable energy reviews*, 6(1-2):3–65.

Barker, A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, 18:119–134.

Beucher, H., Rivoirard, J., Cojan, I., and Ors, F. (2012). Contribution of the process-based modeling to plurigaussian simulation in the case of meandering channelized reservoirs. In *AAPG 2012 Annual Convention & Exhibition*, page 1236143.

Bubnova, A. (2018). *Sur le conditionnement des modèles génétiques de réservoirs chenalisés méandriformes à des données de puits*. PhD thesis, PSL University - MINES ParisTech. Thèse de doctorat, Géosciences et géoingénierie Paris Sciences et Lettres 2018.

Bubnova, A., Ors, F., Rivoirard, J., Cojan, I., and Romary, T. (2020). Automatic determination of sedimentary units from well data. *Mathematical Geosciences*, 52(2):213–231.

Chautru, J. M. (1989). The use of Boolean random functions in geostatistics. In Armstrong, M., editor, *Geostatistics*, pages 201–212, Dordrecht. Springer Netherlands.

Chilès, J. and Delfiner, P. (1999). *Geostatistics: Modeling spatial uncertainty*. Wiley, New York.

Chopin, N. and Papaspiliopoulos, O. (2020). *An introduction to Sequential Monte Carlo*. Springer, Cham.

Cojan, I., Ors, F., and Rivoirard, J. (2012). New method for inferring process-based models to well data in the case of meandering systems. In *AAPG annual Convention & Exhibition in Long Beach*, page 1235352.

Collet, A., Regnault, O., Ozhogin, A., Imantayeva, A., and Garnier, L. (2022). Three-dimensional reactive transport simulation of uranium in situ recovery: Large-scale well field applications in Shu Saryssu Bassin, Tortkuduk deposit (Kazakhstan). *Hydrometallurgy*, 211:105873.

Daams, R., Díaz-Molina, M., and Mas, R. (1996). Uncertainties in the stratigraphic analysis of fluvial deposits from the Loranca Basin, Central Spain. *Sedimentary Geology*, 102(1-2):187–209.

Davis, W. M. (1903). I.—the development of river meanders. *Geological Magazine*, 10(4):145–148.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Probability and Its Applications. Springer, New York.

Devroye, L. (1986). *Non-uniform random variable generation*. Springer-Verlag, New York.

Díaz Molina, M., Arribas Mocoroa, J., and Bustillo, M. (1989). The Tortola and Villalba de la Sierra fluvial fans, late Oligocene-Early Miocene. Loranca Basin, Central Spain. *Conference: 4th International Conference on fluvial sedimentology. Fieldtrip 7,Guidebook*.

Díaz-Molina, M., Marzo, M., and Puigdefábregas, C. (1993). Geometry and lateral accretion patterns in meander loops: examples from the upper oligocene–lower miocene, Loranca basin, Spain. In *Alluvial sedimentation*, volume 17, pages 115–131. International Association of Sedimentologists.

Douc, R., Cappé, O., and Moulines, É. (2005). Comparison of resampling schemes for particle filtering. *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.*, pages 64–69.

Doucet, A., Freitas, N., and Gordon, N. (2001). An introduction to sequential Monte Carlo methods. In *Sequential Monte Carlo Methods in Practice*, pages 3–13. Springer-Verlag, New York.

Dubrule, O. (1989). A review of stochastic models for petroleum reservoirs. *Geostatistics*, pages 493–506.

Dupont, E., Zhang, T., Tilke, P., Liang, L., and Bailey, W. (2018). Generating realistic geology conditioned on physical measurements with generative adversarial networks. *arXiv preprint arXiv:1802.03065*.

Dépret, T., Riquier, J., and Piégay, H. (2017). Evolution of abandoned channels: Insights on controlling factors in a multi-pressure river system. *Geomorphology*, 294:99–118.

Earle, S. (2016). *Physical geology*. Campus Manitoba.

Emery, X. (2007). Simulation of geological domains using the plurigaussian model: new developments and computer programs. *Computers & geosciences*, 33(9):1189–1201.

Flumy-Userguide (2022). FLUMY: Process-based channelized reservoir models. Free download from: https://flumy.minesparis.psl.eu.

Freulon, X. (2013). Simulation of truncated Gaussian vectors by genetic algorithm. Journées de Géostatistique.

Freulon, X. and de Fouquet, C. (1993). Conditioning a Gaussian model with inequalities. In Soares, A., editor, *Geostatistics Tróia'92*, pages 201–212. Kluwer,Dordrecht.

Friedkin, J. F. (1945). *A laboratory study of the meandering of alluvial rivers.* United States Waterways Experiment Station.

Gay, G. R., Gay, H. H., Gay, W. H., Martinson, H. A., Meade, R. H., and Moody, J. A. (1998). Evolution of cutoffs across meander necks in Powder River, Montana, USA. *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Group*, 23(7):651–662.

Gedler, G. (1991). Algorithme de simulation du schéma Booléen conditionnel. Technical report, Ecole des Mines de Paris, Centre de Géostatistique.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:721–741.

Greco, A., Jeulin, D., and Serra, J. (1979). The use of the texture analyser to study sinter structure: application to the morphology of calcium ferrites encountered in basic sinters of rich iron ores. *Journal of Microscopy*, 116(2):199–211.

Grimaud, J.-L., Ors, F., Lemay, M., Cojan, I., and Rivoirard, J. (2022). Preservation and completeness of fluvial meandering deposits influenced by channel motions and overbank sedimentation. *Journal of Geophysical Research: Earth Surface*, page e2021JF006435.

Haldorsen, H. and MacDonald, C. (1987). Stochastic modeling of underground reservoir facies (smurf). In *Spe annual technical conference and exhibition.* OnePetro.

Hall, P. (1988). *Introduction to the theory of the coverage processes.* Wiley, New York.

Harms, J. C., Mackenzie, D. B., and McCubbin, D. G. (1963). Stratification in modern sands of the Red River, Louisiana. *The Journal of Geology*, 71:566 – 580.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57(1):97–109.

Hooke, J. (1995). River channel adjustment to meander cutoffs on the river Bollin and river Dane, northwest England. *Geomorphology*, 14(3):235–253.

Ikeda, S., Parker, G., and Sawai, K. (1981). Bend theory of river meanders. part 1. linear development. *Journal of Fluid Mechanics*, 112:363–377.

Jackson II, R. G. (1976). Largescale ripples of the lower wabash river. *Sedimentology*, 23(5):593–623.

Jeulin, D. (2021). *Morphological models of Random Structures.* Springer.

Johannesson, H. and Parker, G. (2013). Linear theory of river meanders. In *River Meandering*, chapter 7, pages 181–213. American Geophysical Union (AGU).

Kingman, J. F. C. (1992). *Poisson processes*, volume 3. Clarendon Press.

Kroese, D. P., Taimre, T., and Botev, Z. I. (2013). *Handbook of Monte Carlo methods.* John Wiley & Sons.

Lake, L. W. (1989). *Enhanced oil recovery.* Old Tappan, NJ; Prentice Hall Inc.

Lantuéjoul, C. (2002). *Geostatistical Simulation: Models and Algorithms.* Springer, Berlin.

Lantuéjoul, C. (2013). Exact simulation of a Boolean model. *Image Analysis and Stereology*, 32:101–105.

L'Ecuyer, P. (2017). History of uniform random number generation. In *2017 Winter Simulation Conference (WSC)*, pages 202–230.

Lemay, M. (2018). *Transposition à l'environnement turbiditique chenalisé d'un modèle de systèmes fluviatiles méandriformes pour la modélisation de réservoirs*. PhD thesis, PSL University - MINES ParisTech. Thèse de doctorat, Géosciences et géoingénierie Paris Sciences et Lettres 2018.

Leopold, L. B. and Wolman, M. G. (1957). *River channel patterns: braided, meandering, and straight*. US Government Printing Office.

Leopold, L. B. and Wolman, M. G. (1960). River meanders. *GSA Bulletin*, 71(6):769–793.

Lewis, G. and Lewin, J. (1983). Alluvial cutoffs in Wales and the Borderlands. *Modern and ancient fluvial systems*, pages 145–154.

Lopez, S. (2003). *Channelized Reservoir Modeling: a Stochastic Process-based Approach*. Theses, École Nationale Supérieure des Mines de Paris.

Lopez, S., Cojan, I., Rivoirard, J., and Galli, A. (2009). Process-based stochastic modelling: Meandering channelized reservoirs. In *Analogue and Numerical Modelling of Sedimentary Systems: From Understanding to Prediction*, pages 139–144. John Wiley & Sons, Ltd.

Marbut, C. (1896). *The Physical Features of Missouri*. Tribune printing Company, state printers.

Matheron, G. (1967). *Eléments pour une théorie des milieux poreux*. Masson, Paris.

Matheron, G. (1975). *Random Sets and Integral Geometry*. John Wiley & Sons, New York.

Matheron, G. (2019). *Matheron's Theory of Regionalised Variables*. Oxford University Press.

Metropolis, N., Rosebluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091.

Meyn, S. and Tweedie, R. (1993). *Markov chain and stochastic stability*. Springer, London.

Molchanov, I. (1997). *Statistics of the Boolean model for practioners and mathematicians*. Wiley, Chichester.

Morgensen, K. and Tweedie, R. (1996). Rates of convergence of the Hastings and Metropolis algoritms. *The Annals of Statistics*, 24-1:101–121.

Pingping, S., Xinwei, L., and Qiujie, L. (2009). Methodology for estimation of CO_2 storage capacity in reservoirs. *Petroleum Exploration and Development*, 36(2):216–220.

Pradhan, A., Kumar Khatua, K., and Sankalp, S. (2018). Variation of velocity distribution in rough meandering channels. *Advances in Civil Engineering*, 2018.

Robert, C. and Casella, G. (2004). *Monte Carlo Statistical Methods*. Springer, New York.

Roberts, G. and Smith, A. (1994). Simple conditions for the Gibbs sampler and Metropolis-Hastings algorithm. *Stochastic Processes and their Applications*, 49:207–216.

Rubinstein, R. (1981). *Simulation and the Monte Carlo method*. John Wiley, New York.

Sanz-Alonso, D., Stuart, A., and Taeb, A. (2018). Inverse problems and data assimilation. *arXiv:1810.06191*.

Sun, T., Meakin, P., and Jøssang, T. (2001). A computer model for meandering rivers with multiple bed load sediment sizes: 2. computer simulations. *Water Resources Research*, 37(8):2243–2258.

Tierney, L. (1994). Markov Chains for exploring posterior distributions. *The Annals of Statistics*, 22-4:1701–1762.

Troncoso, A., Emery, X., Cáceres, A., and Daisy, A. (2017). Application of the Gibbs sampler to the conditional simulation of geological domains. In *Geomine - Mineplanning 2017*, Santiago, Chili. 5th International Seminar on Geology for the Mining Industry – 5th International Seminar on Mine Planning, Oral exposition.

Troncoso, A., Freulon, X., and Lantuéjoul, C. (2022). Sequential simulation of a conditional Boolean model. *Mathematical Geosciences*, 54(2):389–411.

van Dijk, W. M., van de Lageweg, W. I., and Kleinhans, M. G. (2013). Formation of a cohesive floodplain in a dynamic experimental meandering river. *Earth Surface Processes and Landforms*, 38(13):1550–1565.

Walker, R. G. (1976). Facies model - 3. Sandy Fluvial Systems. *Geoscience Canada*, 3(2).

Weill, P., Cojan, I., Beucher, H., Rivoirard, J., and Ors, F. (2013). Process-based modelling for a finer characterization of meandering fluvial reservoir heterogeneities: FLUMY and the Miocene Loranca Basin. In *10th International Conference on Fluvial Sedimentology*, pages 446–447, Leeds, United Kingdom.

Åke Sundborg (1956). The river Klarälven a study of fluvial processes. *Geografiska Annaler*, 38(2-3):125–316.

RÉSUMÉ

Une méthode séquentielle de Monte Carlo, appelée filtrage particulaire, a été utilisée dans un contexte spatial pour simuler deux modèles de réservoir en respectant les faciès observés aux puits. Le premier modèle, le schéma Booléen, est un modèle à base d'objets. Il peut servir à modéliser des réservoirs à deux faciès: un faciès poreux, et un faciès imperméable qui agit comme barrière à la circulation des fluides. Ce modèle se prête bien à des calculs mathématiques : il existe des méthodes statistiques pour en inférer les paramètres, ainsi qu'un algorithme itératif de simulation conditionnelle. Cependant, la vitesse de convergence de cet algorithme est difficile à établir. Un algorithme séquentiel basé sur un filtre particulaire est proposé en alternative. Au final, cet algorithme séquentiel surpasse l'algorithme itératif en termes de qualité des résultats et de temps de calcul.
Le second modèle, Flumy, est un modèle de processus sédimentaires, permettant de représenter la formation de systèmes chenalisés méandriformes. Ce modèle permet de reproduire l'hétérogénéité induite par les géométries complexes des dépots sédimentaires. L'algorithme courant implémenté dans Flumy modifie dynamiquement les processus au cours du temps pour s'adapter aux données et permet ainsi d'obtenir des simulations conditionnelles. Le développement de cet algorithme de conditionnement requiert, cependant, une profonde compréhension de ces processus pour les modifier tout en évitant les artefacts ou les biais. Pour cette raison, un autre algorithme, dit séquentiel, a été développé. Il consiste à construire le réservoir par empilement de couches horizontales au moyen d'un filtre particulaire permettant d'assimiler les facies observés en chaque couche. Ces deux algorithmes de conditionnement ont été comparés sur un cas synthétique et sur un cas réel (bassin de Loranca, en Espagne). Ils fournissent des résultats comparables, mais se distinguent en termes de ressources nécessaires pour leur mise en œuvre: l'agorithme séquentiel requiert une puissance de calcul informatique conséquente quand l'algorithme dynamique nécessite une fine compréhension des processus à modifier.

MOTS CLÉS

Modèles de réservoir; simulation conditionnelle; simulation à base d'objets; simulation à base de processus; simulation séquentielle, filtrage particulaire

ABSTRACT

A sequential Monte Carlo method, called particle filtering, has been used in a spatial context to produce simulations of two reservoir models that reproduce the observed facies at wells. The first one, the Boolean model, is an object-based model. It can be used to model two-facies reservoirs: one porous facies, and an impermeable facies that acts as a barrier for the fluid circulation. The model is mathematically tractable: there exists statistical methods to infer its parameters as well as an iterative conditional simulation algorithm. However, the convergence rate of this algorithm is difficult to establish. A sequential algorithm based on the particle filtering is proposed as an alternative. It finally appears that this sequential algorithm outperforms the iterative algorithm in terms of quality of results and computational time.
The second model, Flumy, is a model of sedimentary processes. It is used for representing the formation of meandering channelized systems. This model can reproduce the heterogeneity induced by the complex geometries of sedimentary deposits. The current algorithm implemented in Flumy modifies dynamically the processes for fitting the data at best to produce conditional simulations. The set-up of this algorithm requires a deep knowledge of the processes to modify them and avoid artifacts and biases. For this reason, another conditioning algorithm, called sequential, has been developed. It consists in building the reservoir by stacking horizontal layers using particle filtering, thus allowing the observed facies to be assimilated in each layer. These two algorithms have been compared on a synthetic case and on a real case (Loranca Basin, Spain). Both give comparable results, but they differ in terms of the resources required for their implementation: whereas the sequential algorithm needs high computer power, the dynamic algorithm requires a fine understanding of the processes to be modified.

KEYWORDS

Reservoir models; conditional simulation; object-based model; process-based model; sequential simulation; particle filtering