

# GML6402A : Géostatistique

## Cours 8 : Simulation de faciès



**POLYTECHNIQUE  
MONTREAL**

UNIVERSITÉ  
D'INGÉNIERIE

# Objectifs

- Décrire le fonctionnement de méthodes de simulation de faciès (simulation d'indicatrice, gaussien tronqué, plurigaussien tronqué, Gibbs, multipoints);
- Décrire les avantages et inconvénients et limitations de chaque méthode;
- Associer les covariances d'indicatrices à des drapeaux de codage;
- Associer des images simulées à des drapeaux de codage;
- Décrire la méthode de Gibbs pour obtenir des réalisations conditionnelles aux faciès observés.



# Plan du cours

1. Mise en contexte
2. Simulation de faciès
  - Simulation d'indicatrices
  - Simulation gaussienne tronquée
  - Simulation plurigaussienne
3. Conditionnement aux faciès observées
  - Échantillonneur de Gibbs
  - Calage par S-STBM
4. Méthode multipoint (début)
  - SNESIM
  - Approches séquentielles par paquet (Filtersim, simpat, parchwork,...)

# 1. Mise en contexte

## Objectifs de la simulation de faciès :

Quantifier l'incertitude et les risques géologiques du réservoir pétrolier ou du gisement

- Fournir aux ingénieurs pétroliers et miniers des représentations de la distribution spatiale de l'hétérogénéité du réservoir;
- Réaliser des simulations d'écoulement ( ou de récupération) sur les simulations ( utilisant les propriétés du modèle) pour optimiser le développement du champ.

# 1. Mise en contexte

## Contexte :

On cherche à modéliser la géologie (faciès) des réservoirs pétroliers et des gisements. ( Aussi applicable à des modèles hydrogéologiques)

### Réservoir pétrolier :

Construire un modèle maillé du réservoir contenant ses propriétés pétrophysiques (la porosité, les perméabilités, la pression capillaire...) afin de simuler son comportement lors de la production.

### Gisement :

Construire un modèle maillé du gisement contenant les propriétés minéralogiques, l'intensité des fracturations, la génération d'acide, la densité de la roche afin de simuler les comportements géologiques, géotechniques et géométallurgiques.



# 1. Mise en contexte

## Objectifs de simulation de faciès :

### Étape de développement :

- Construire un modèle des propriétés géologiques sur une grille très fine (quasi ponctuel);
- Estimer les réserves et quantifier leur incertitude;
- Définir des scénarios pour le processus de récupération;
- Optimiser la localisation des puits / forages.

# 1. Mise en contexte

## Objectifs de simulation de faciès :

### Début d'exploitation :

- Continuer à intégrer les informations géologiques disponibles (levés sismiques, interprétation des forages);
- Déterminer la connectivité du réservoir ou la continuité spatiale du gisement;
- Concorder la taille de la grille avec le processus de récupération choisi;
- Optimiser la localisation des puits / forages.

# 1. Mise en contexte

## Objectifs de simulation de faciès :

### Conception finale :

- Étendre le modèle des propriétés géologiques sur grille fine vers une grille grossière pour le simulateur d'écoulement ;
- Assure la cohérence de la simulation d'un point de vue géologique;
- Rendre le processus de récupération plus efficace;
- Ajustement du modèle aux données de production (Calage, *history matching*) ;
- Optimiser les forages pour géométrie complexe



## 2. Simulations de faciès

### Méthodes possibles :

- I. Simulation séquentielle d'indicatrices } Basé sur la SGS
- II. Simulation gaussienne tronquée } Basé sur la
- III. Simulations plurigaussiennes } troncation de
- IV. Échantillonneur de Gibbs } champs gaussiens
- V. Calage des faciès observés } Basé sur le
- VI. Méthode multipoint } conditionnement
- } aux faciès observés
- } Inférer les propriétés d'une
- } image d'entraînement aux
- } simulations



## 2.1 Simulation séquentielle d'indicatrices

### Algorithme:

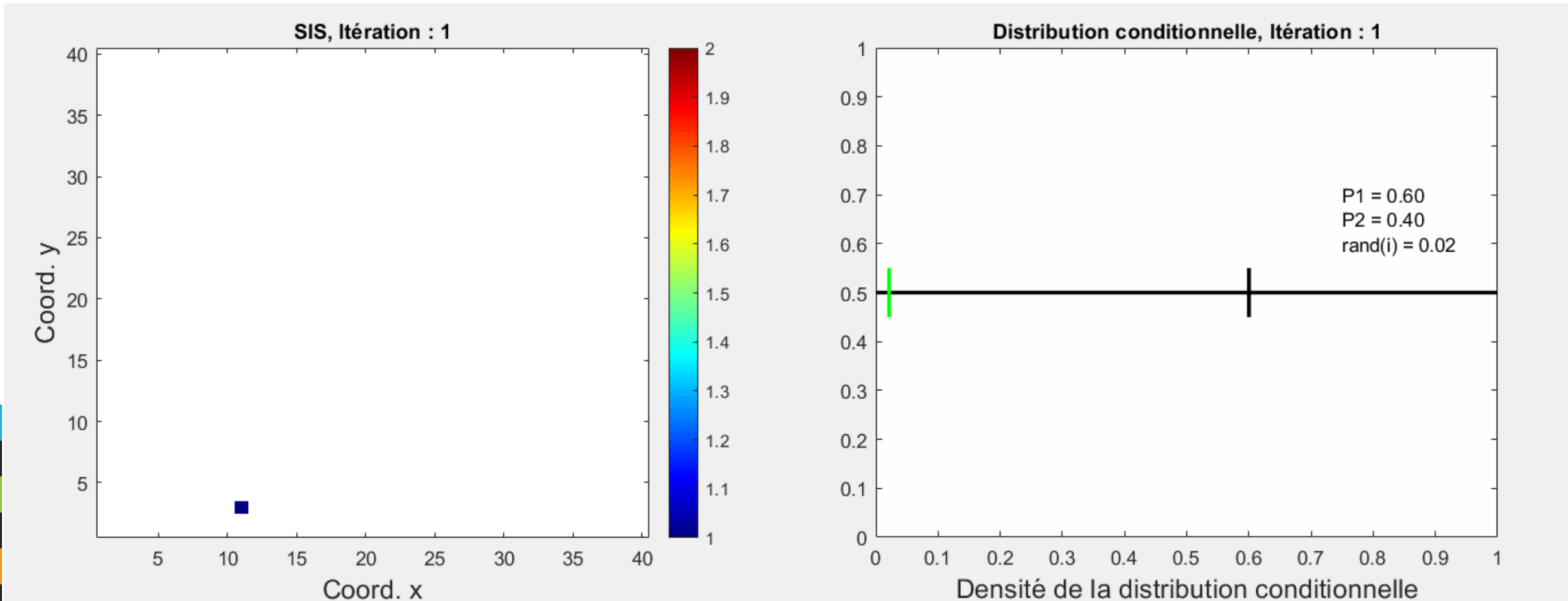
Soit  $Z(x)$  un champ de  $k$  faciès et soit  $Z(x_i), i = 1, \dots, N, N$  faciès observés :

On cherche à simuler  $Z(x_j)$  en  $n$  emplacements  $x_j, j = 1, \dots, n$ , conditionnellement à  $Z(x_i), i = 1, \dots, N$

- a) Coder chaque faciès  $k$  par une indicatrice différente ;
- b) Choisir un point  $x_j$  aléatoirement et kriger les  $k$  faciès en ce point conditionnellement aux données déjà simulées et observées  $\rightarrow p_i, i = 1, \dots, k$ ;
- c) Normaliser les probabilités  $p_i$  (t.q.  $\sum_{i=1}^k p_i = 1$  et  $p_i \geq 0, \forall i$ );
- d) Tirer une valeur aléatoire de la loi  $U(0,1)$  ce qui détermine le faciès au point  $x_j$ ;
- e) Introduire le point  $x_j$  aux données simulées et refaire le processus pour tous les points  $x_j$  restants (retour à l'étape b).

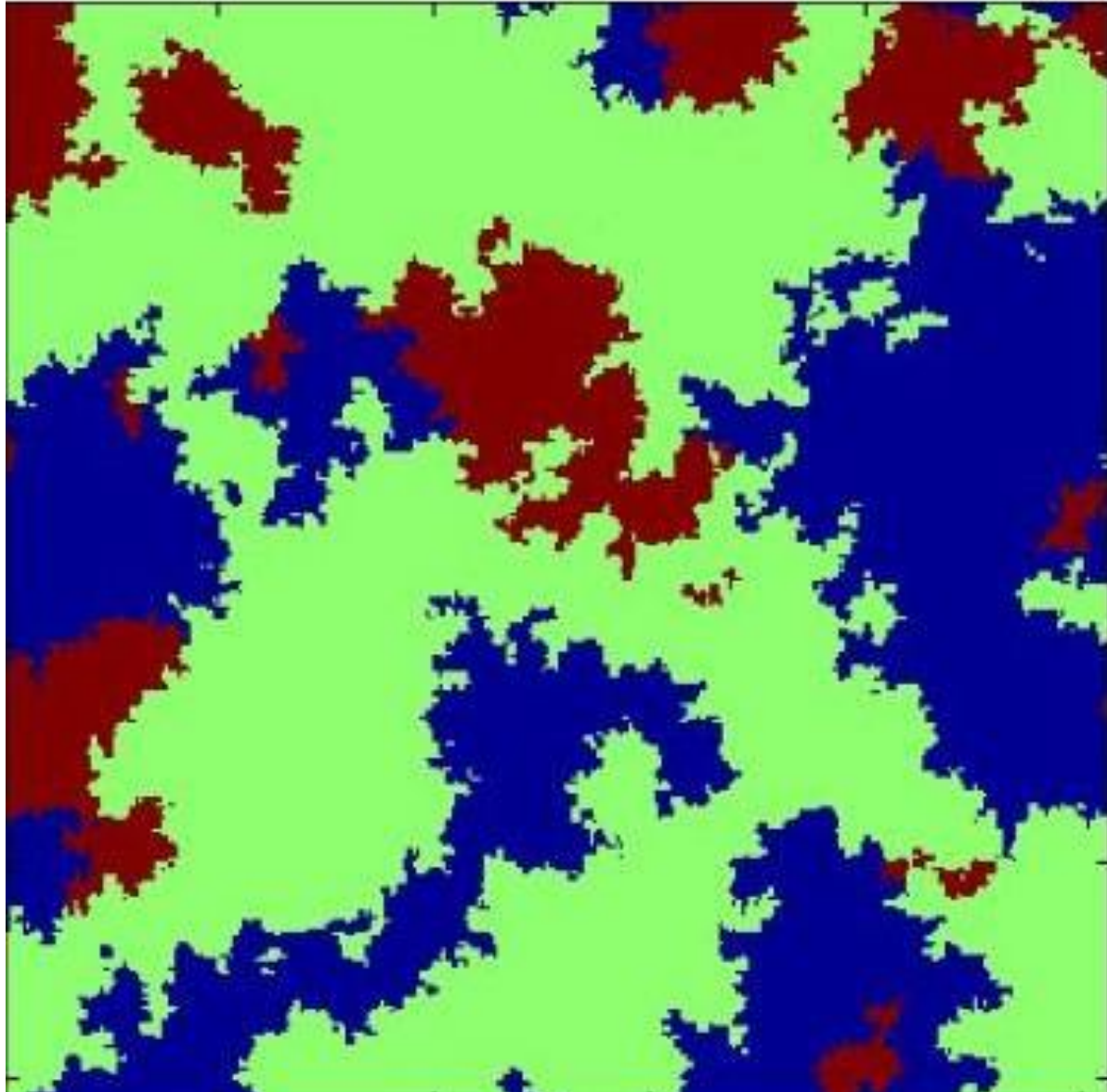
# 2.1 Simulation séquentielle d'indicateurs

**Animation** : simulation d'un champ 40x40 avec un modèle sphérique isotrope ( $a=10$ ) pour les deux variogrammes d'indicateurs. Les proportions sont 0.6 pour le faciès 1 et 0.4 pour le faciès 2.



# 2.1 Simulation séquentielle d'indicatrices

Caractéristiques :



**Vos observations ?**

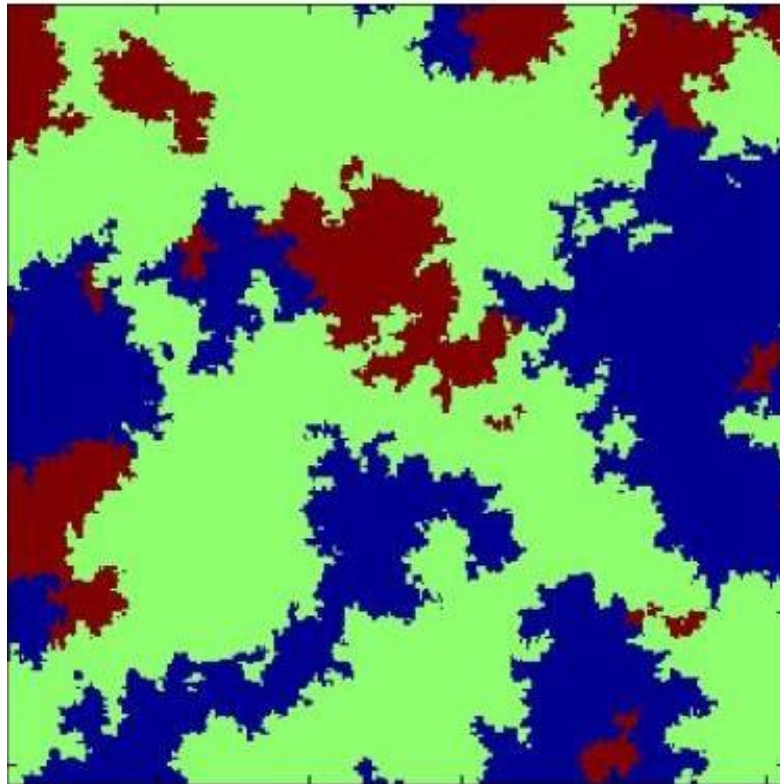
simulation avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$   
variog. sphérique avec  $\alpha = 50$

Bleu :  $p_1$ ; vert :  $p_2$  ; Brun :  $p_3$

## 2.1 Simulation séquentielle d'indicateurs

### Caractéristiques :

- Spatialement, **toutes les transitions** entre faciès sont **possibles**;
- Reproduit les **variogrammes d'indicateurs**, mais pas les covariances croisées entre les indicateurs (on ne les utilise pas).



Toutes les transitions possibles sont observées dans les deux sens et dans toutes les directions.

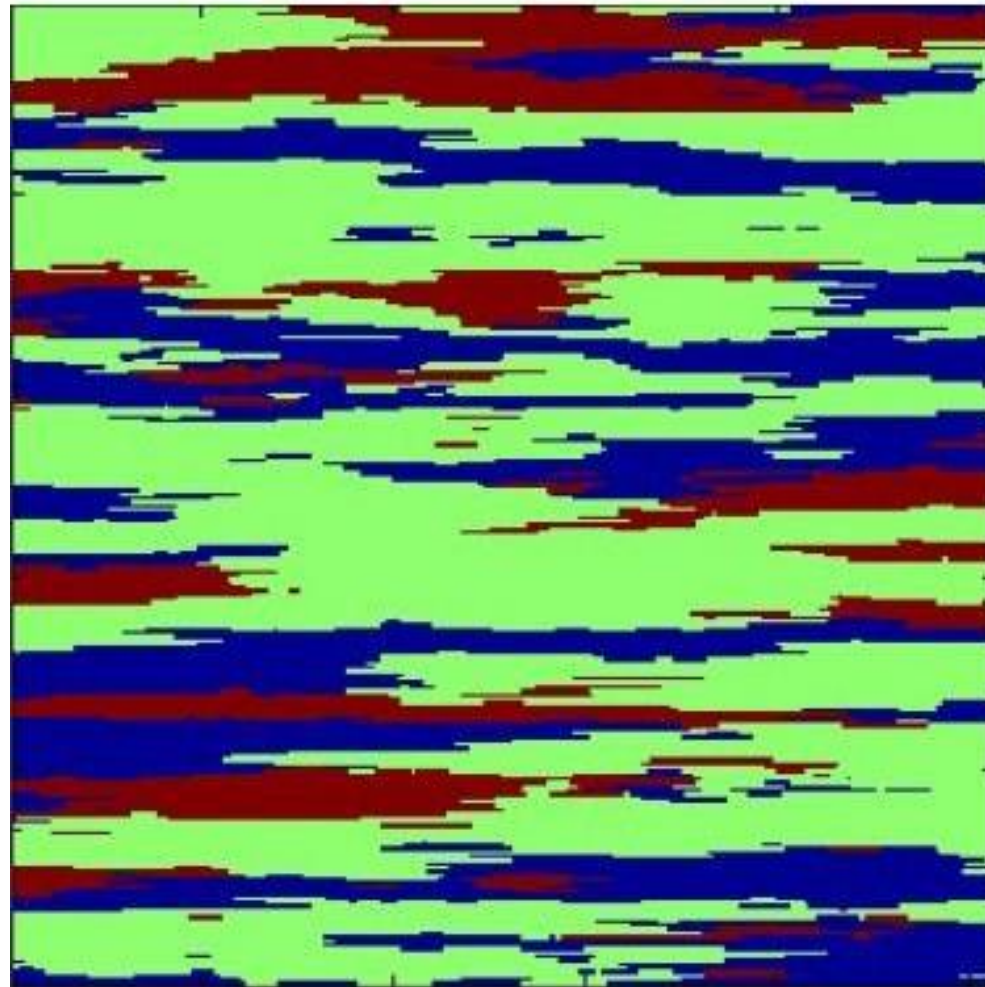
simulation avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$ ,  
variog. sphérique avec  $a = 50$

Bleu :  $p_1$ ; vert :  $p_2$  ; Brun :  $p_3$

# 2.1 Simulation séquentielle d'indicatrices

## Caractéristiques :

- Un peu plus réaliste d'un cas sédimentaire, mais **néanmoins toutes les transitions sont possibles**



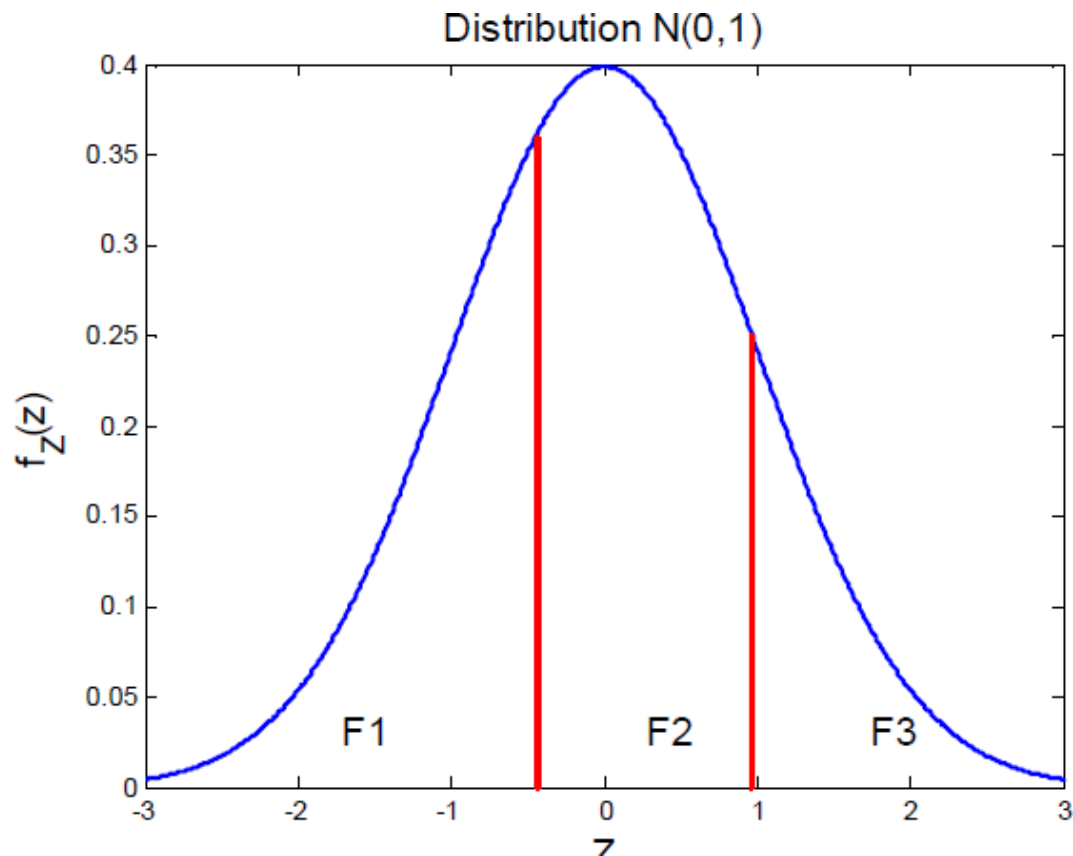
1  
4

Sphérique avec anisotropie  
( $a_x = 200, a_y = 10$ )

## 2.2 Simulation gaussienne tronquée

Idée :

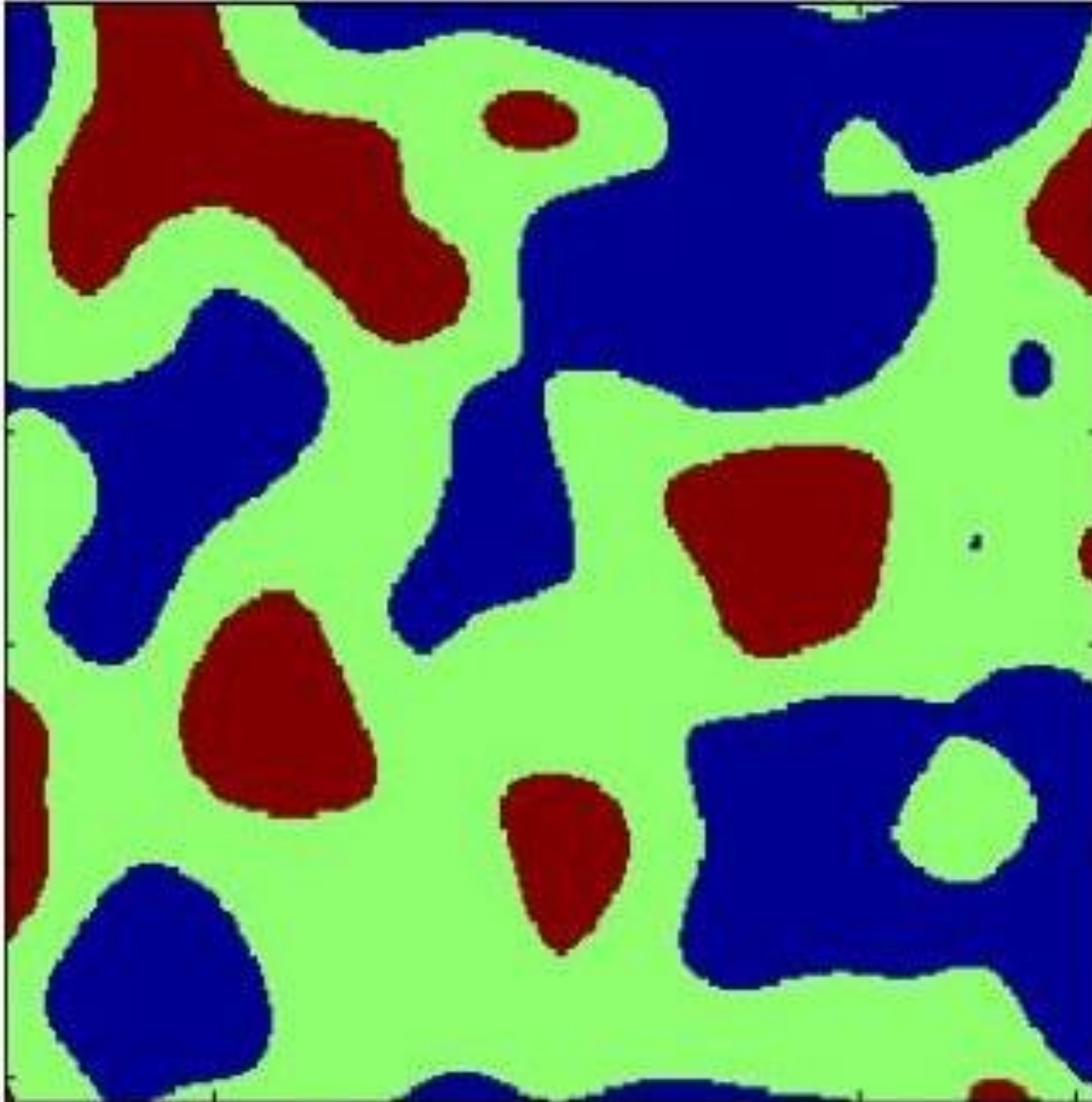
Seuiller la distribution d'un champ gaussien ou chaque seuil définit la frontière entre deux faciès et les proportions de chaque faciès



avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$

## 2.2 Simulation gaussienne tronquée

Visuellement :



**Vos observations ?**

Champ gaussien  
Variogramme gaussien ( $\alpha = 50$ )

Faciès simulés  
avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$

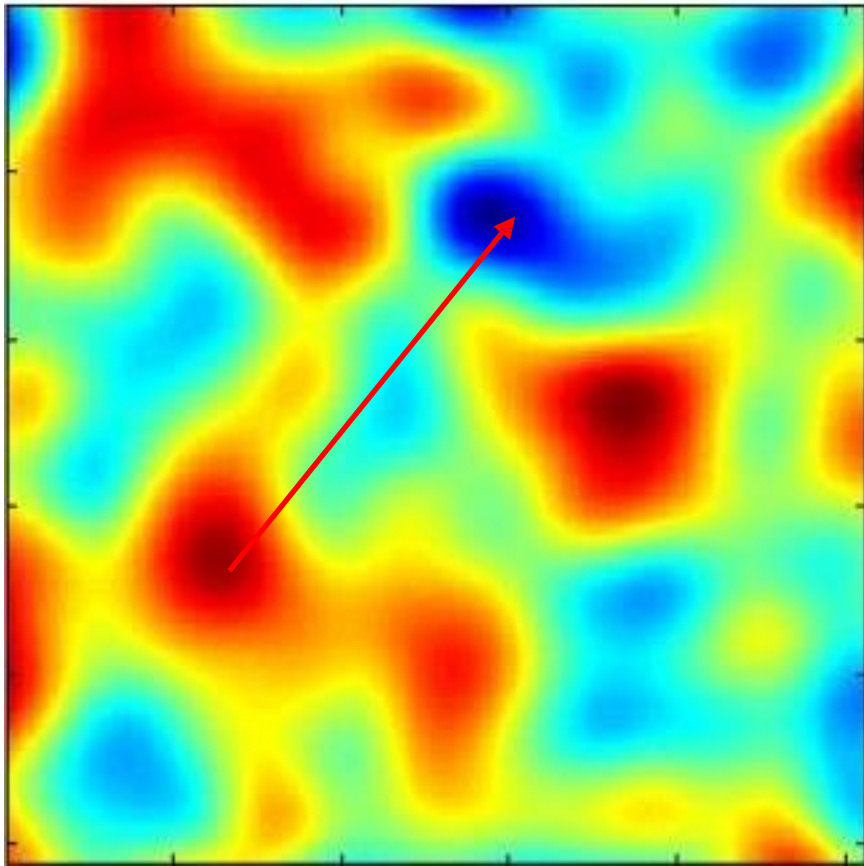
Bleu :  $p_1$ ; vert :  $p_2$  ; Brun :  $p_3$



## 2.2 Simulation gaussienne tronquée

### Visuellement

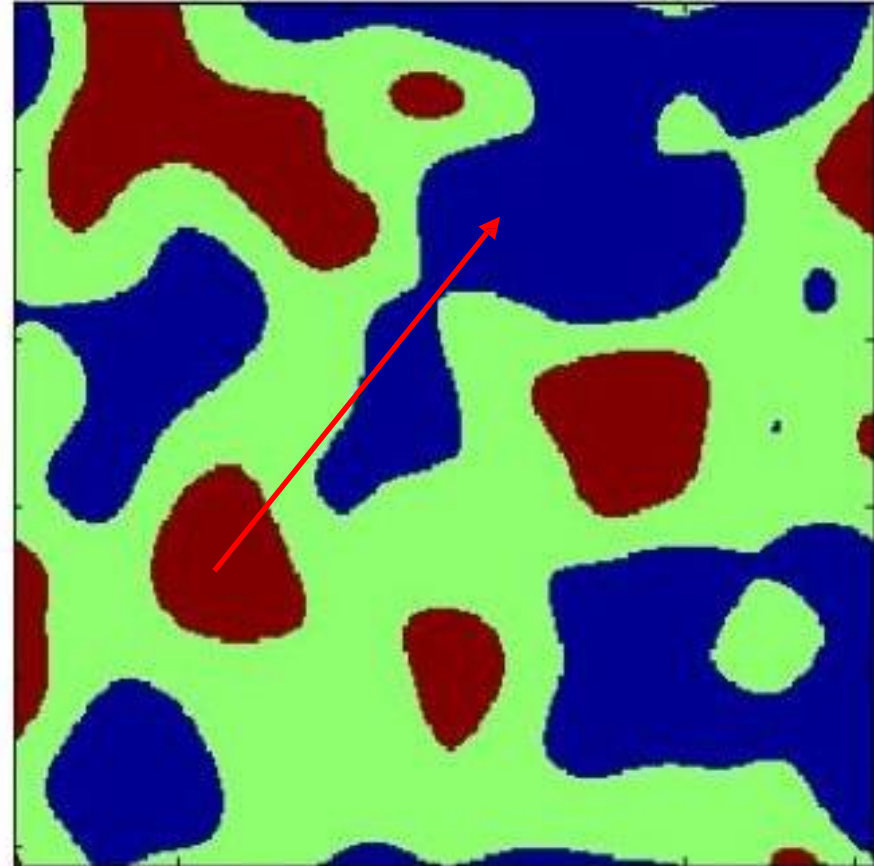
Champ gaussien  
Variogramme gaussien ( $a = 50$ )



Tronqué



Faciès simulés  
avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$



Les faciès brun et bleu ne se touchent jamais. On doit passer par l'intermédiaire en vert.

Bleu :  $p_1$ ; vert :  $p_2$  ; Brun :  $p_3$

## 2.2 Simulation gaussienne tronquée

### Remarque et questions :

Le modèle implique que seuls les faciès successifs peuvent être contigus spatialement. Ainsi la transition  $F1 - F3$  ne peut être observée. Le choix de l'ordre des faciès doit respecter les relations observées.

Deux questions à résoudre:

- a) Comment décider du variogramme de la variable gaussienne?
- b) Comment tenir compte des faciès observés aux points échantillons (quelles valeurs gaussiennes simuler aux points échantillons)?



## 2.2 Simulation gaussienne tronquée

### a) Ajustement du variogramme de la variable gaussienne :

On note que :

$$\begin{aligned} p_{ij}(h) &= E[I_i(x)I_j(x+h)] = P(I_i(x) = 1 \cap I_j(x+h) = 1) \\ &= P(\{c_{i-1} < Z(x) \leq c_i\} \cap \{c_{j-1} < Z(x+h) \leq c_j\}) \end{aligned}$$

$E[I_i(x)I_j(x+h)]$ : calculé à partir des faciès observés

$P(\{c_{i-1} < Z(x) \leq c_i\} \cap \{c_{j-1} < Z(x+h) \leq c_j\})$ : calculé à partir de la loi binormale si on connaît  $C(h)$

## 2.2 Simulation gaussienne tronquée

a) Ajustement du variogramme de la variable gaussienne :

$E[I_i(x)I_j(x+h)]$ : calculé à partir des faciès observés

$P(\{c_{i-1} < Z(x) \leq c_i\} \cap \{c_{j-1} < Z(x+h) \leq c_j\})$ : calculé à partir de la loi binormale si on connaît  $C(h)$

Choisir  $C(h)$  telle que les deux termes <sup>2</sup>soit presque d'égalité pour toutes les distances  $h$ .

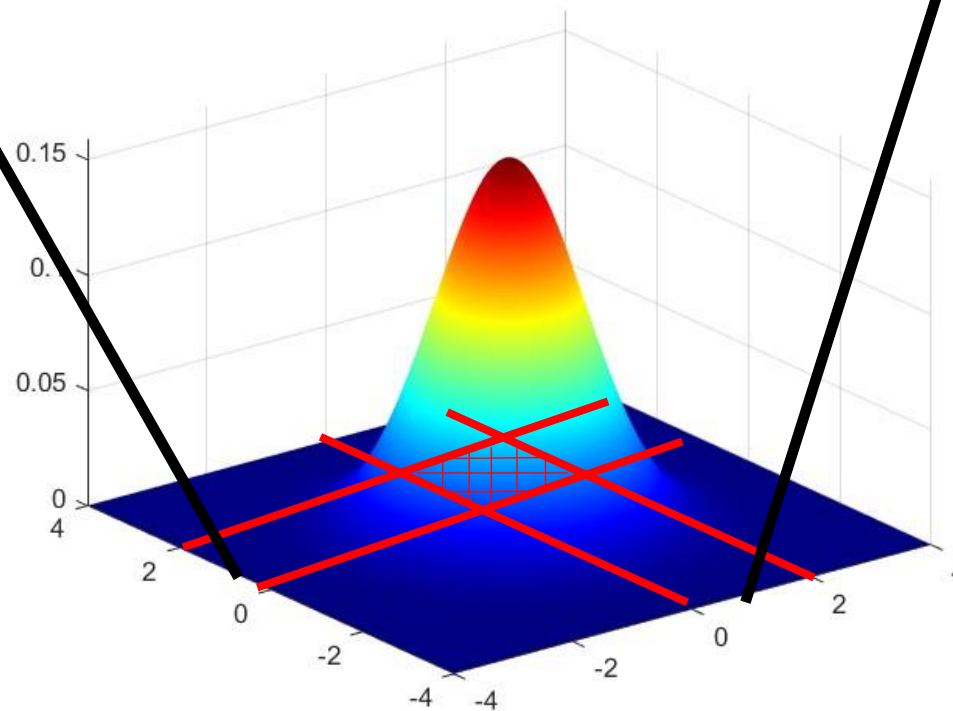
## 2.2 Simulation gaussienne tronquée

a) Ajustement du variogramme de la variable gaussienne :

$$E[I_i(x)I_j(x+h)]$$

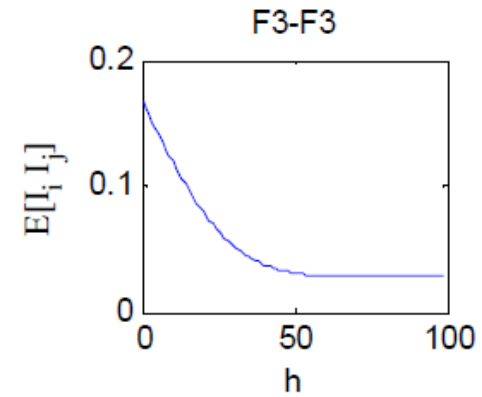
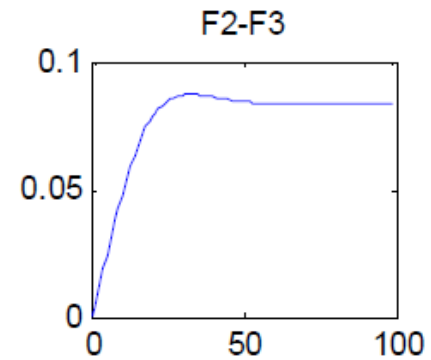
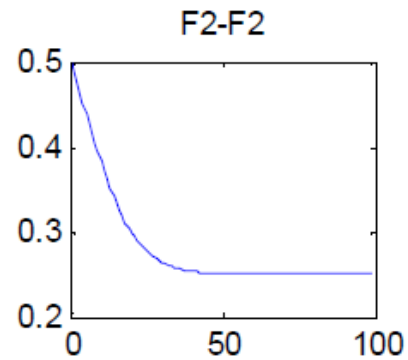
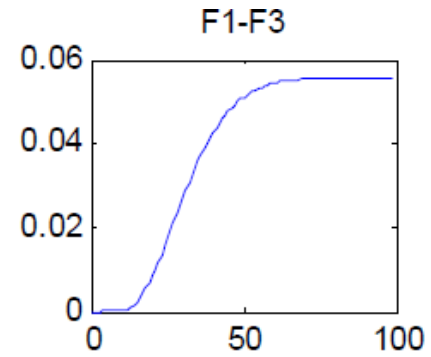
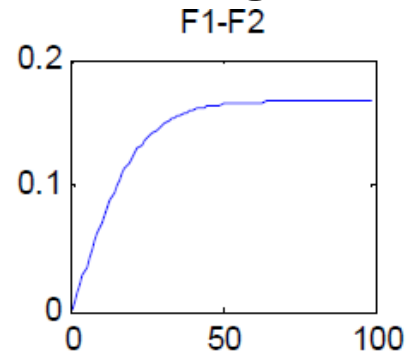
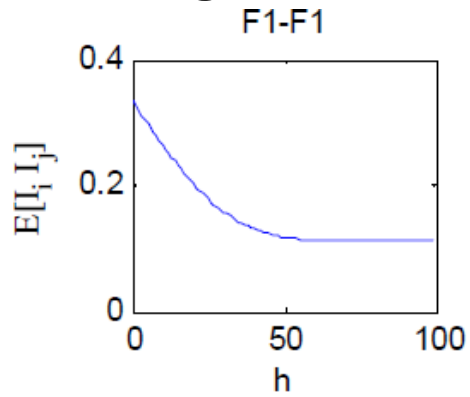
$$P(\{c_{i-1} < Z(x) \leq c_i\} \cap \{c_{j-1} < Z(x+h) \leq c_j\})$$

$$m_1=0, m_2=0, \sigma_1=1, \sigma_2=1, \rho=0$$



# 2.2 Simulation gaussienne tronquée

a) Ajustement du variogramme de la variable gaussienne :



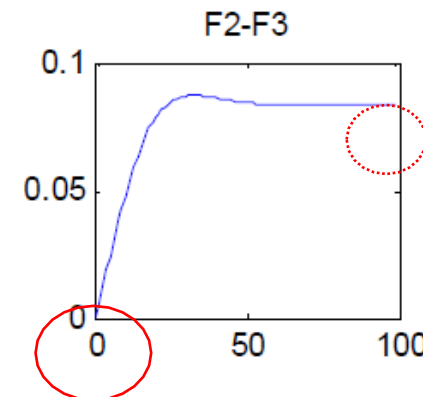
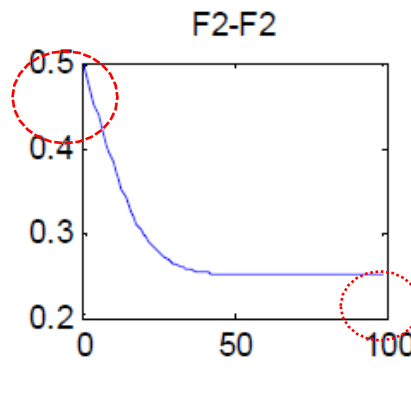
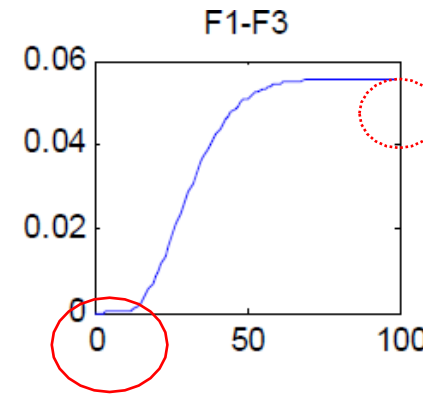
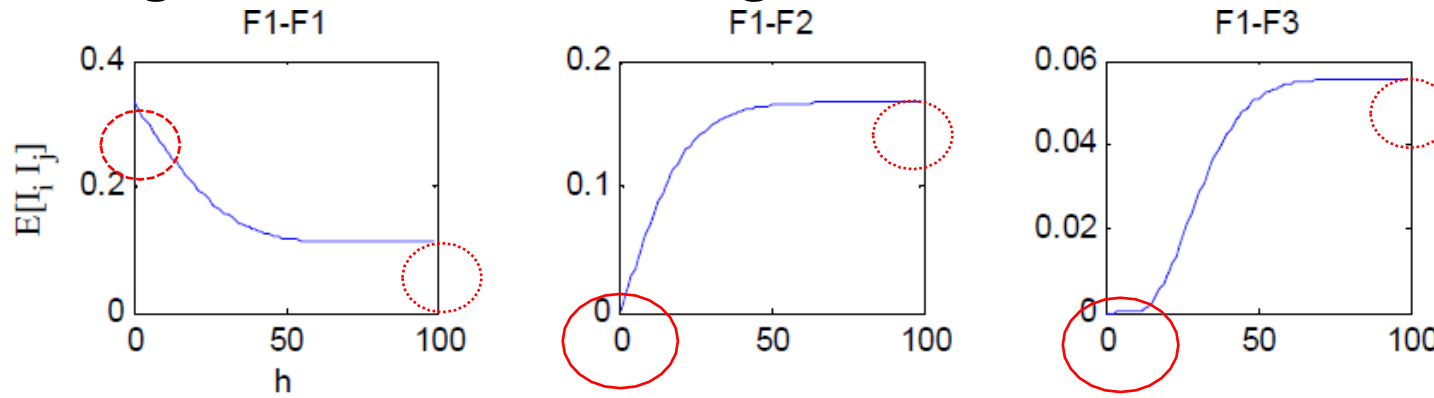
$$p_1 = \frac{1}{3}, p_2 = \frac{1}{2} \text{ et } p_3 = \frac{1}{6}$$

Variogramme gaussien ( $a = 50$ )

**Vos observations ?**

# 2.2 Simulation gaussienne tronquée

a) Ajustement du variogramme de la variable gaussienne :



$$p_1 = \frac{1}{3}, p_2 = \frac{1}{2} \text{ et } p_3 = \frac{1}{6}$$

Variogramme gaussien ( $a = 50$ )

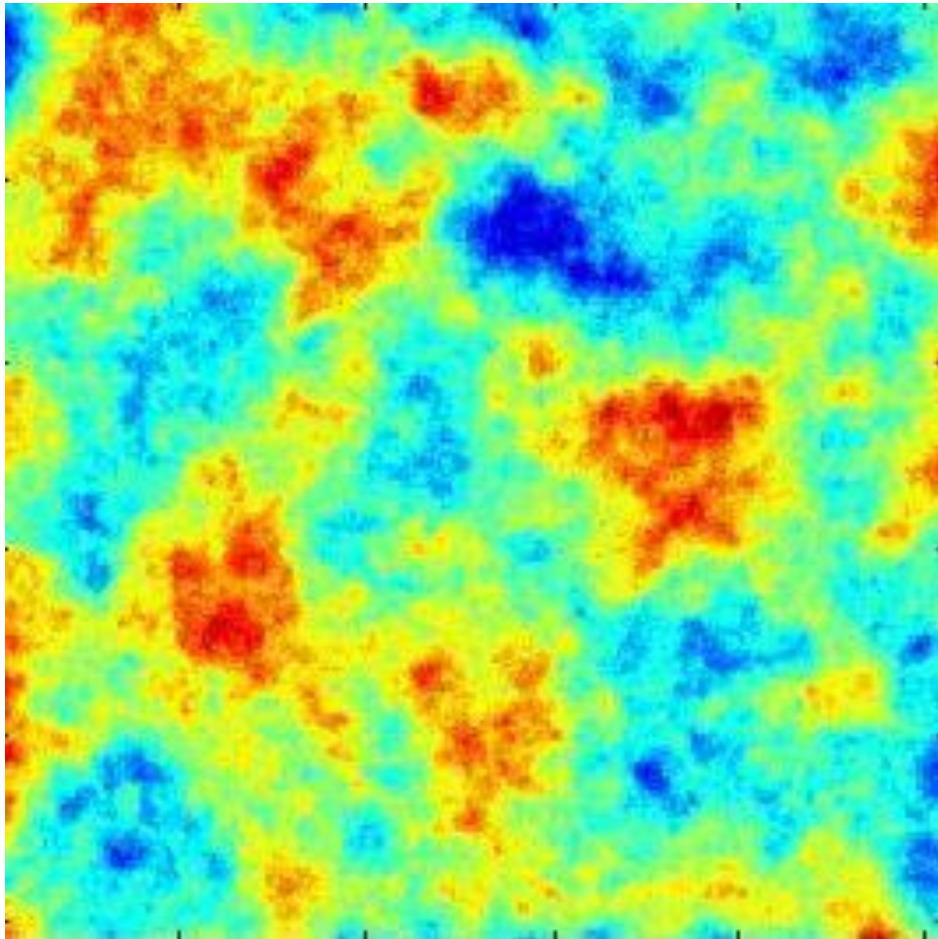
- = 0
- =  $p_i$
- =  $p_i p_j$



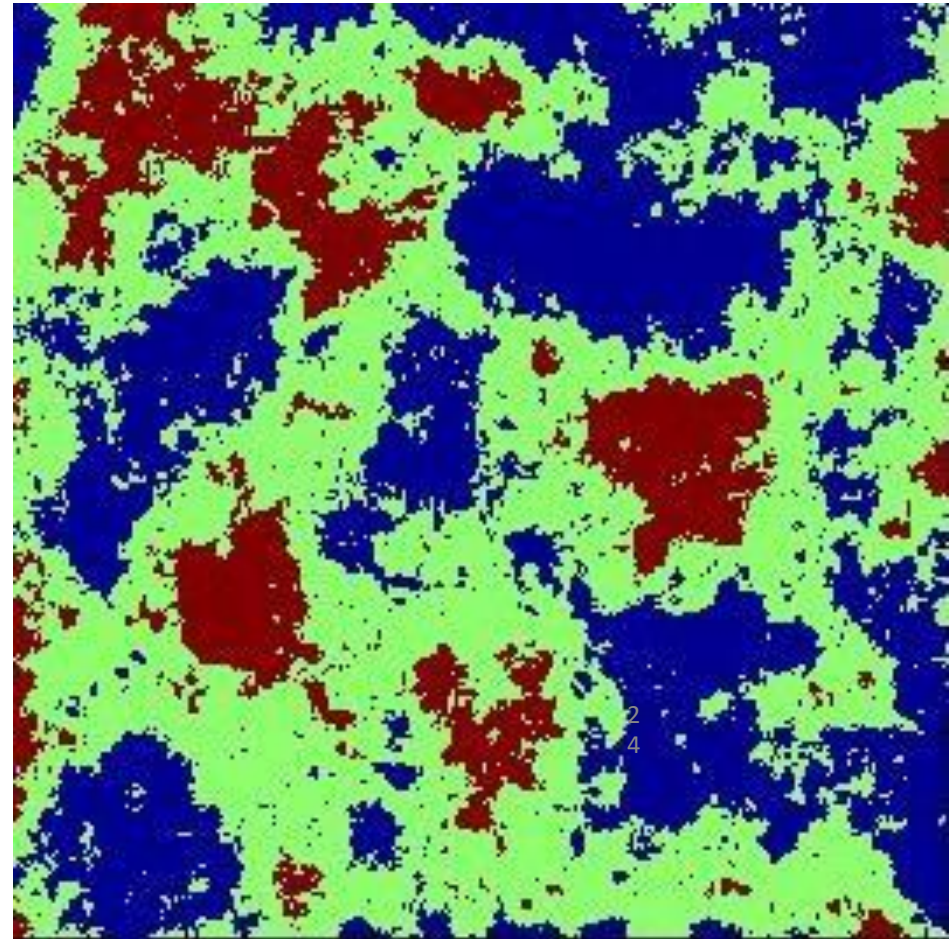
## 2.2 Simulation gaussienne tronquée

### D'autres exemples

Champ gaussien  
Variogramme sphérique ( $a = 50$ )



Faciès simulés  
avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$



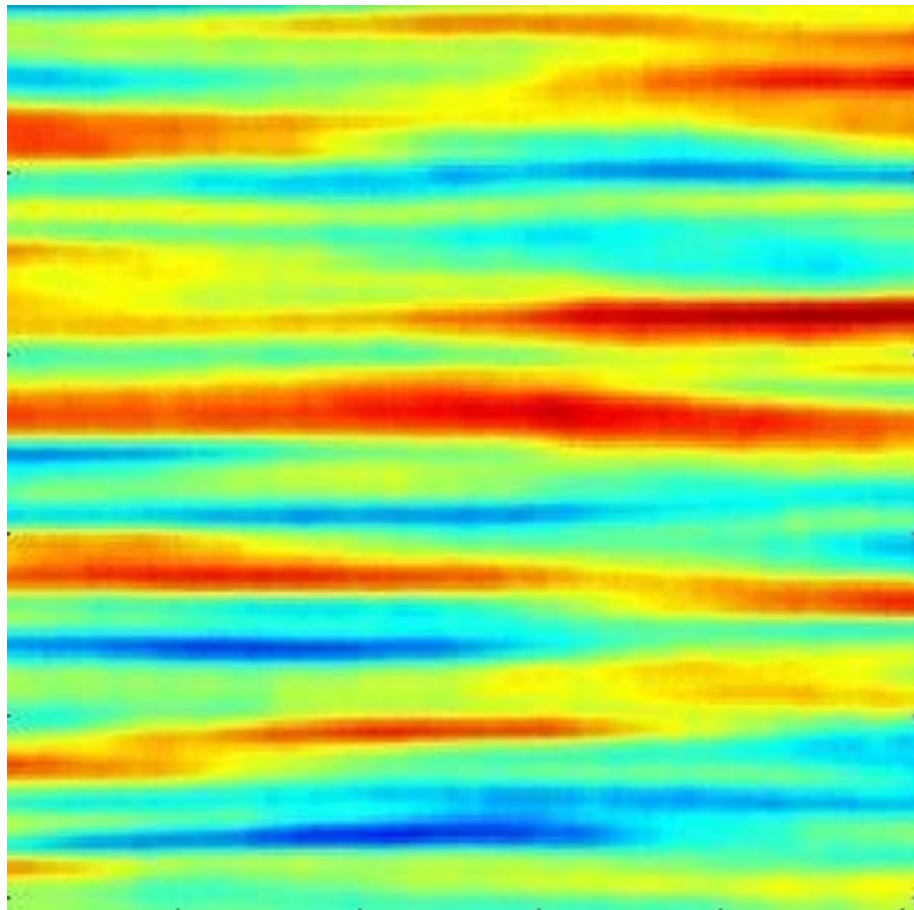
Bleu :  $p_1$  ; vert :  $p_2$  ; Brun :  $p_3$



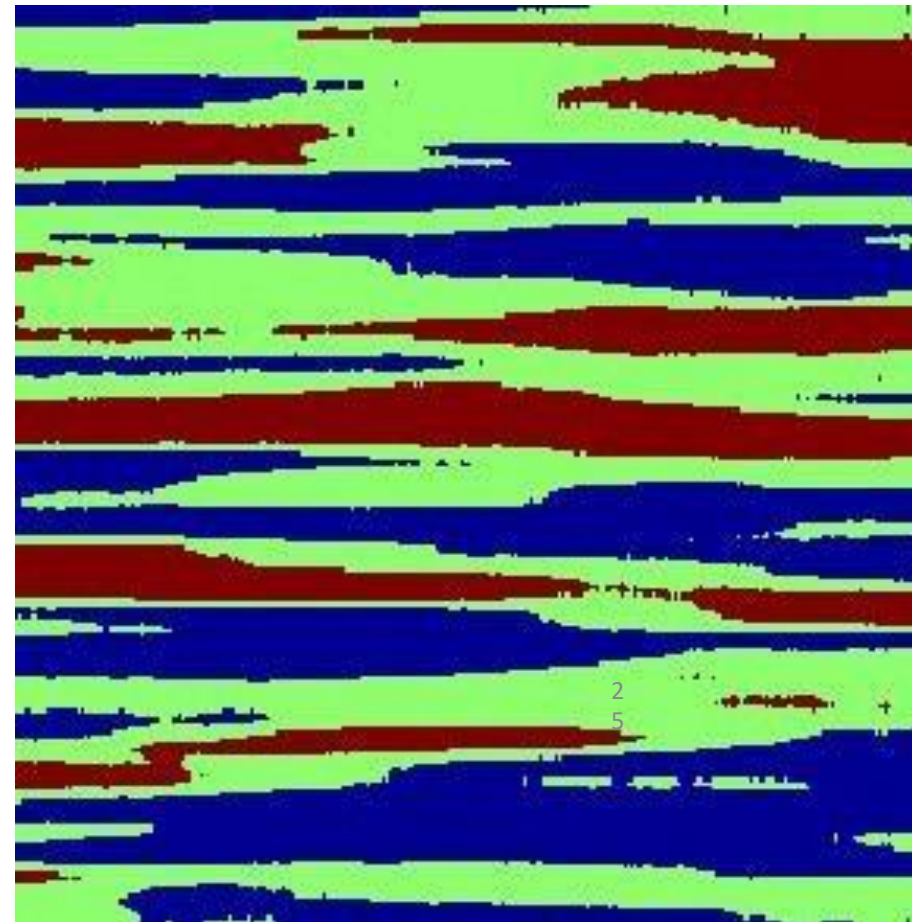
## 2.2 Simulation gaussienne tronquée

### D'autres exemples

Champ gaussien  
Variogramme gaussien anisotrope



Faciès simulés  
avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$

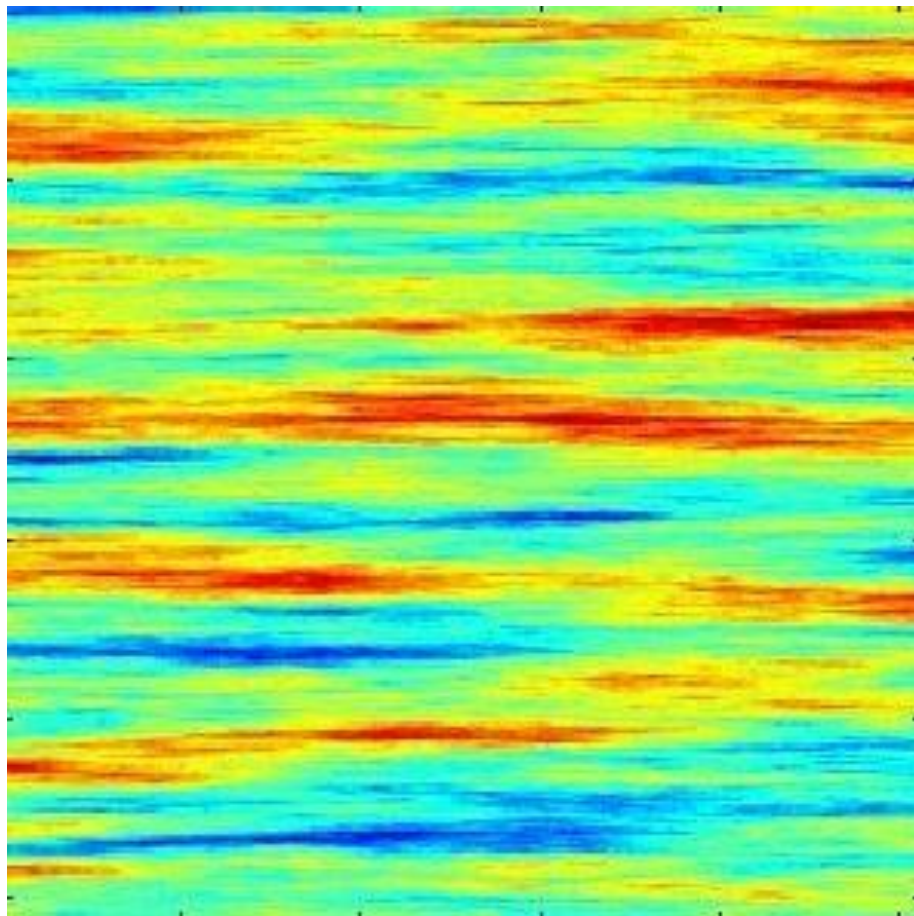


Bleu :  $p_1$ ; vert :  $p_2$ ; Brun :  $p_3$

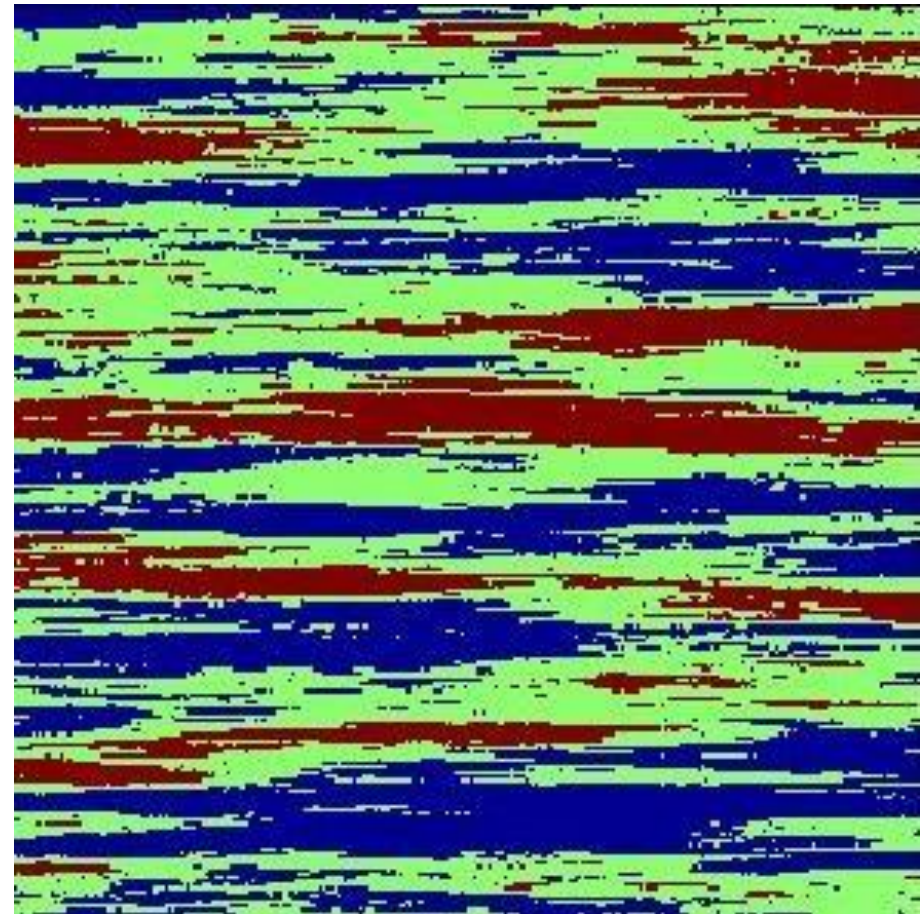
## 2.2 Simulation gaussienne tronquée

### D'autres exemples

Champ gaussien  
Variogramme sphérique anisotrope



Faciès simulés  
avec  $p_1 = \frac{1}{3}$ ,  $p_2 = \frac{1}{2}$  et  $p_3 = \frac{1}{6}$

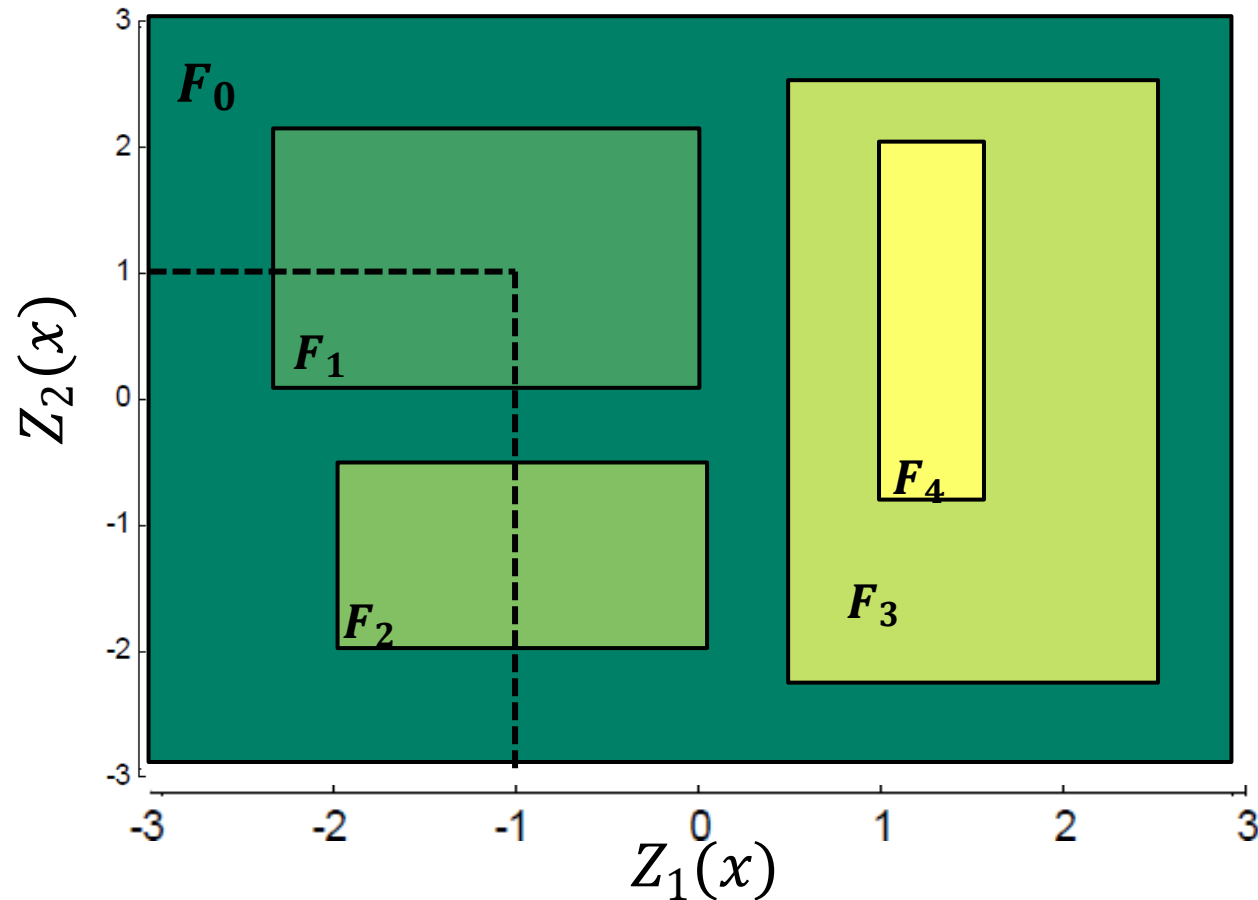


Bleu :  $p_1$ ; vert :  $p_2$  ; Brun :  $p_3$

## 2.3 Simulation plurigaussienne

**Idée :** Généraliser la simulation gaussienne tronquée à plusieurs champs gaussiens et ajouter plus de flexibilité à l'agencement des faciès

Exemple avec deux variables aléatoires gaussiennes :



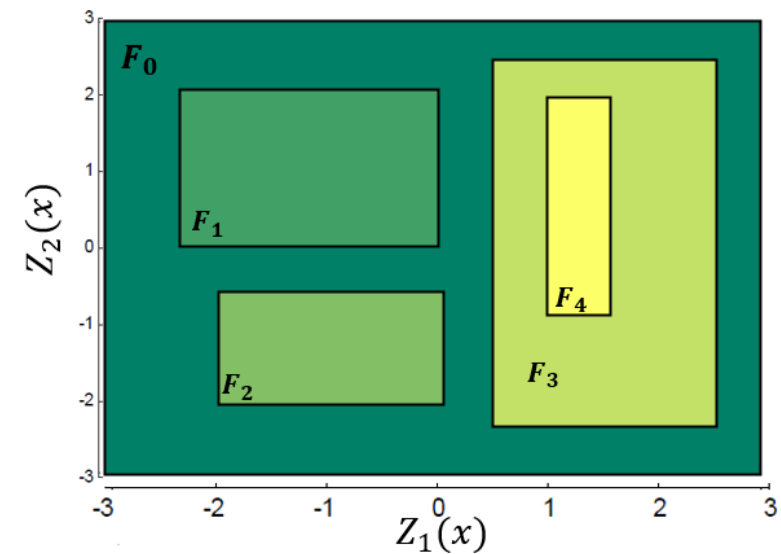
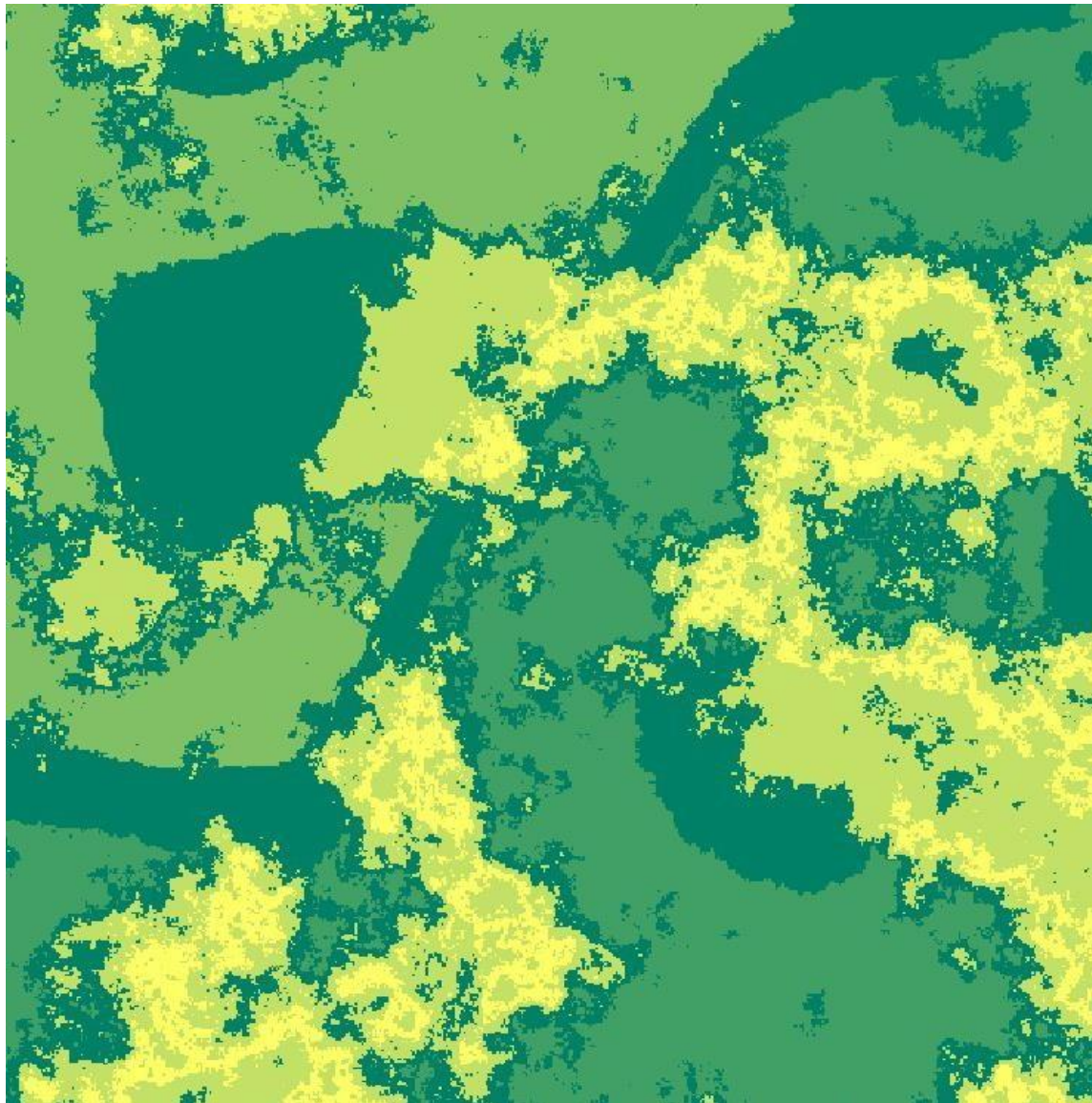
$$Z_1(x_1) = -1$$
$$Z_2(x_1) = 1$$

Donc,

$$I(x_1) = F_1$$

## 2.3 Simulation plurigaussienne

Exemples :



$Z_1$  a un variogramme sphérique  $a = 150$

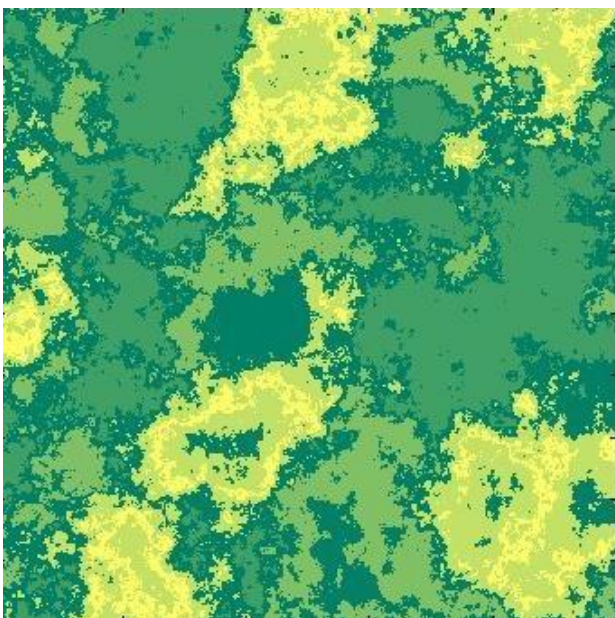
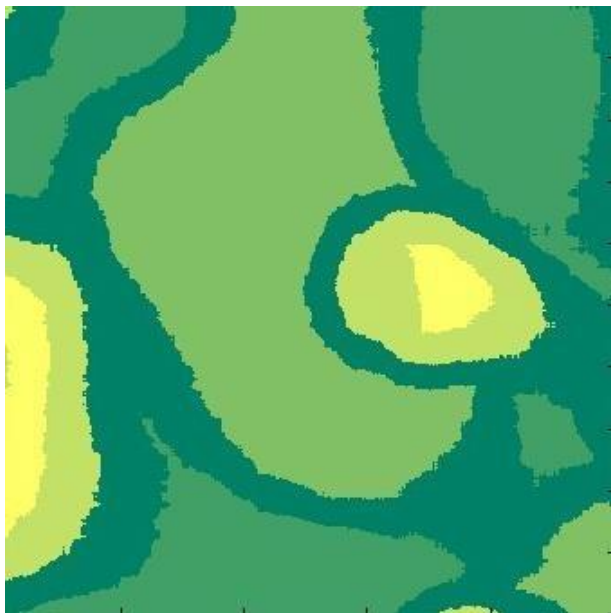
$Z_2$  a un variogramme gaussien  $a = 260$

$\frac{2}{8}$

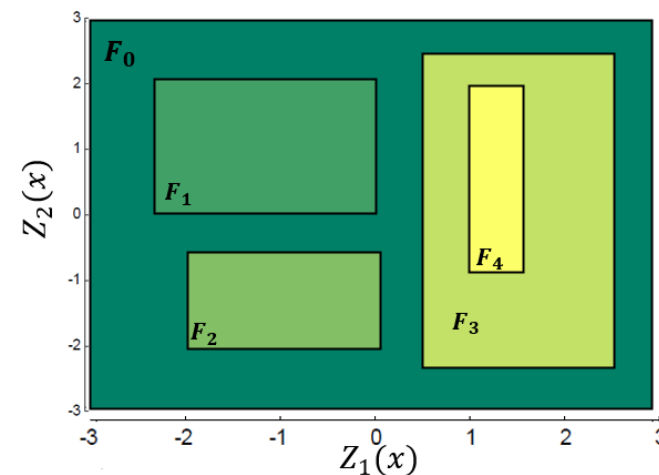
Grille : 500 × 500

## 2.3 Simulation plurigaussienne

Exemples :



Même drapeau, mais  $Z_1$  et  $Z_2$  ont  
Variogramme gaussien avec  $a = 260$

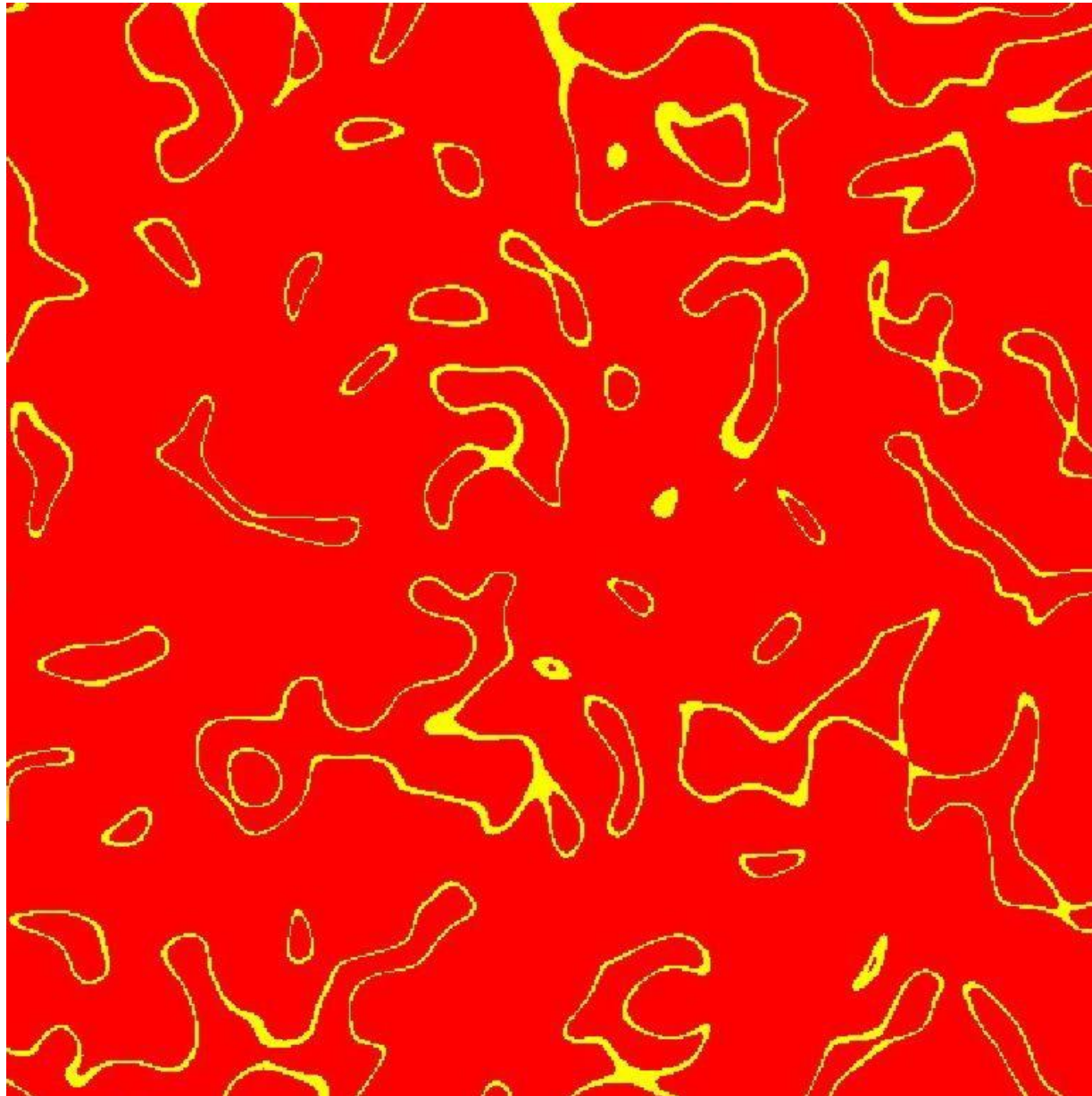


Même drapeau, mais  $Z_1$  et  $Z_2$  ont  
Variogramme sphérique avec  $a = 150$

Grille :  $500 \times 500$

## 2.3 Simulation plurigaussienne

Exemples :



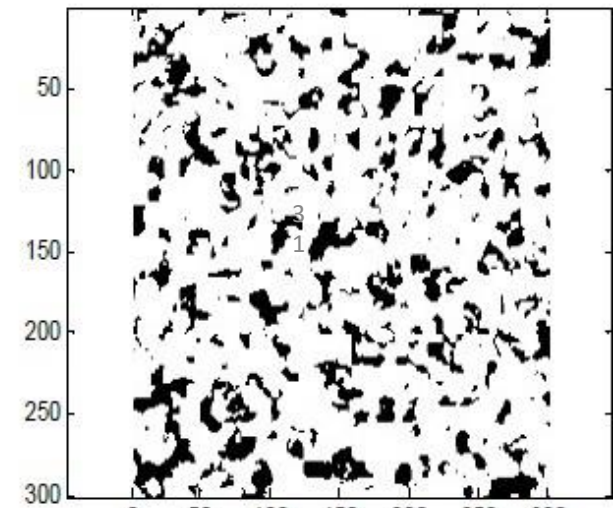
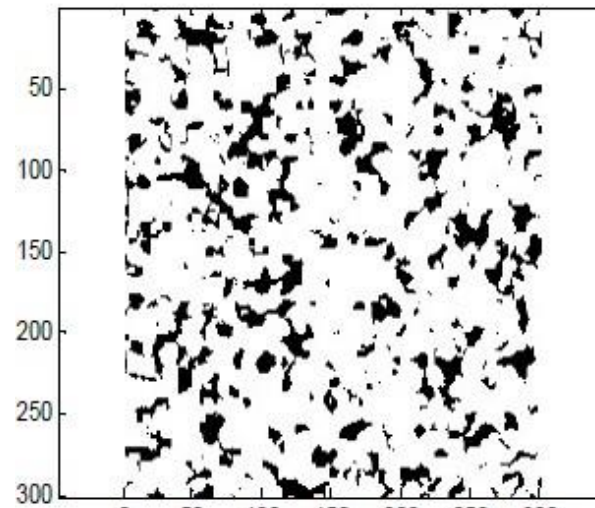
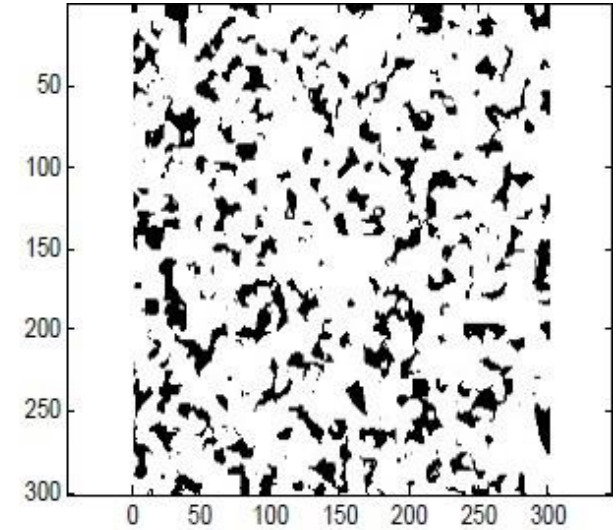
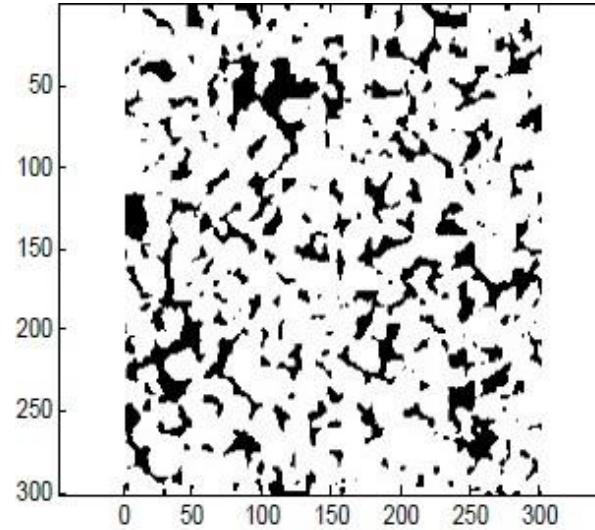
Grille : 500 × 500



## 2.3 Simulation plurigaussienne

Tests : identifier l'image réelle parmi les 4 images

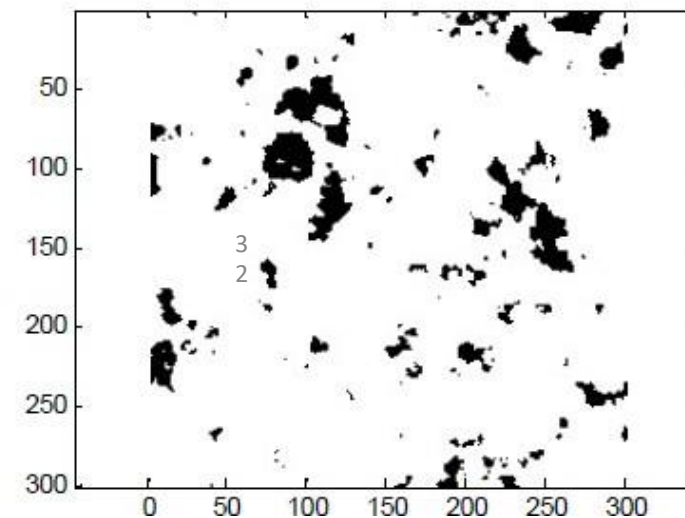
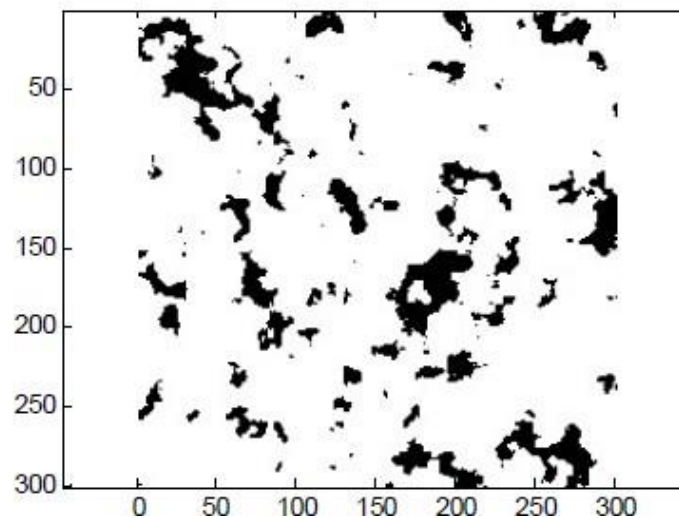
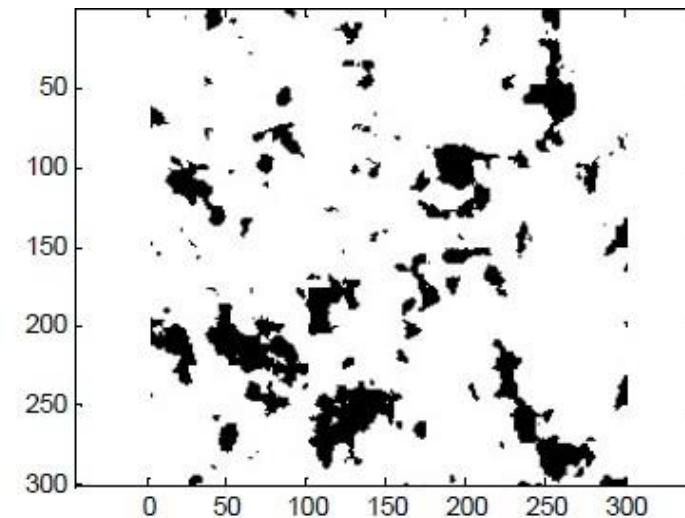
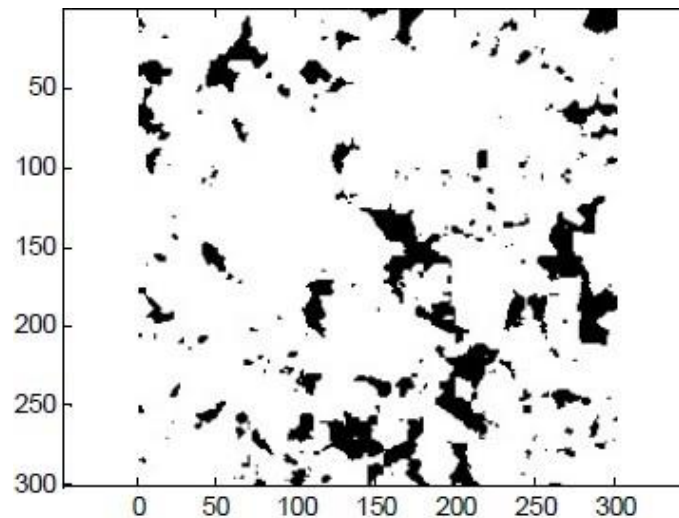
Image réelle : un grès  
Noir : les pores



## 2.3 Simulation plurigaussienne

Tests : identifier l'image réelle parmi les 4 images

Image réelle : un  
carbonate  
Noir : les pores

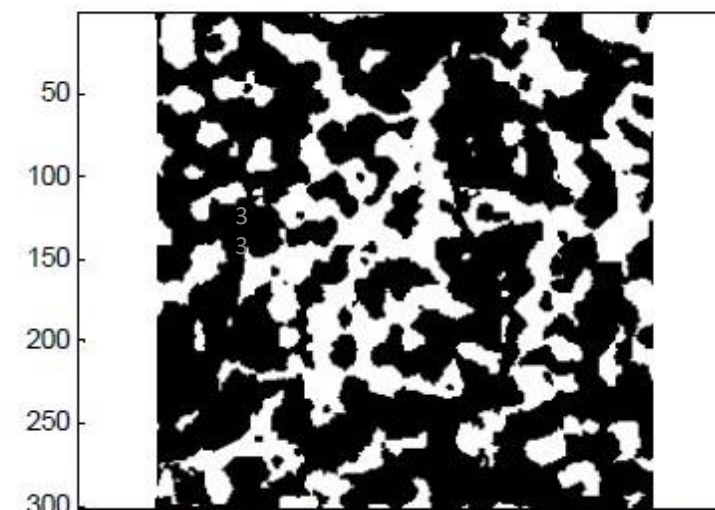
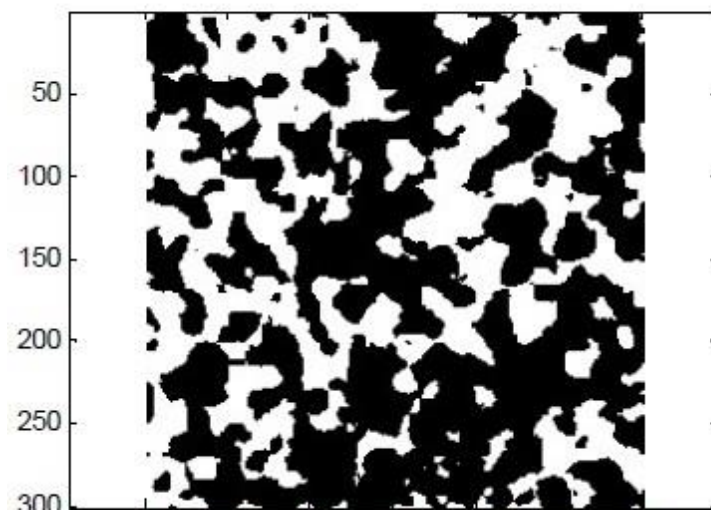
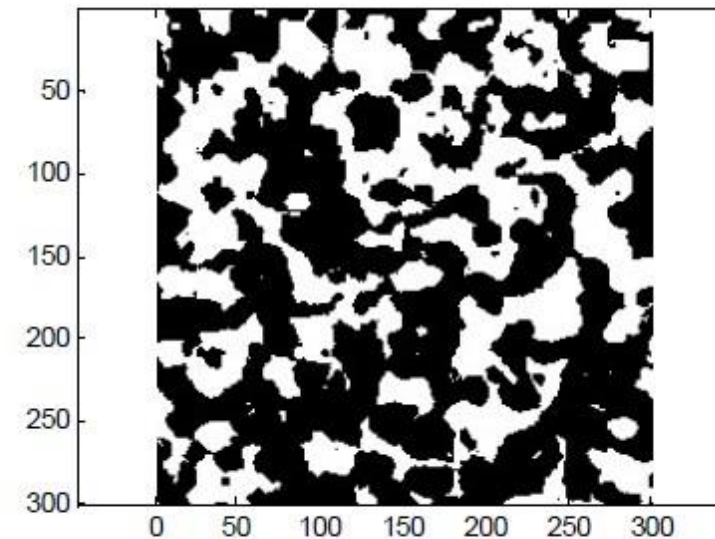
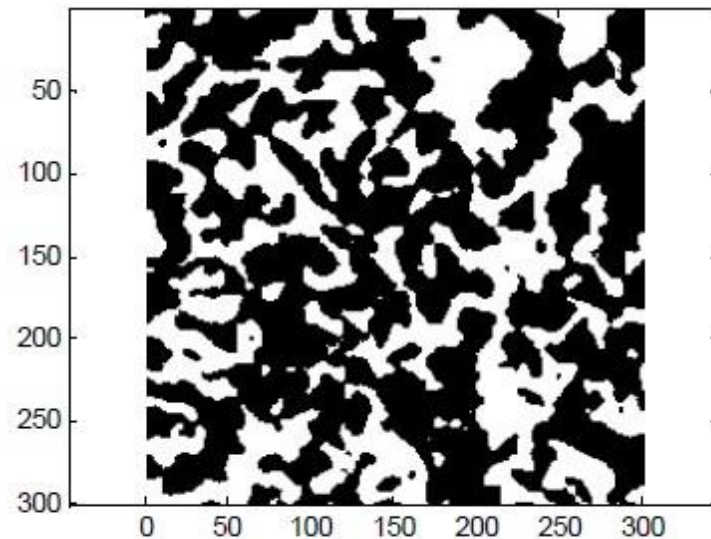




## 2.3 Simulation plurigaussienne

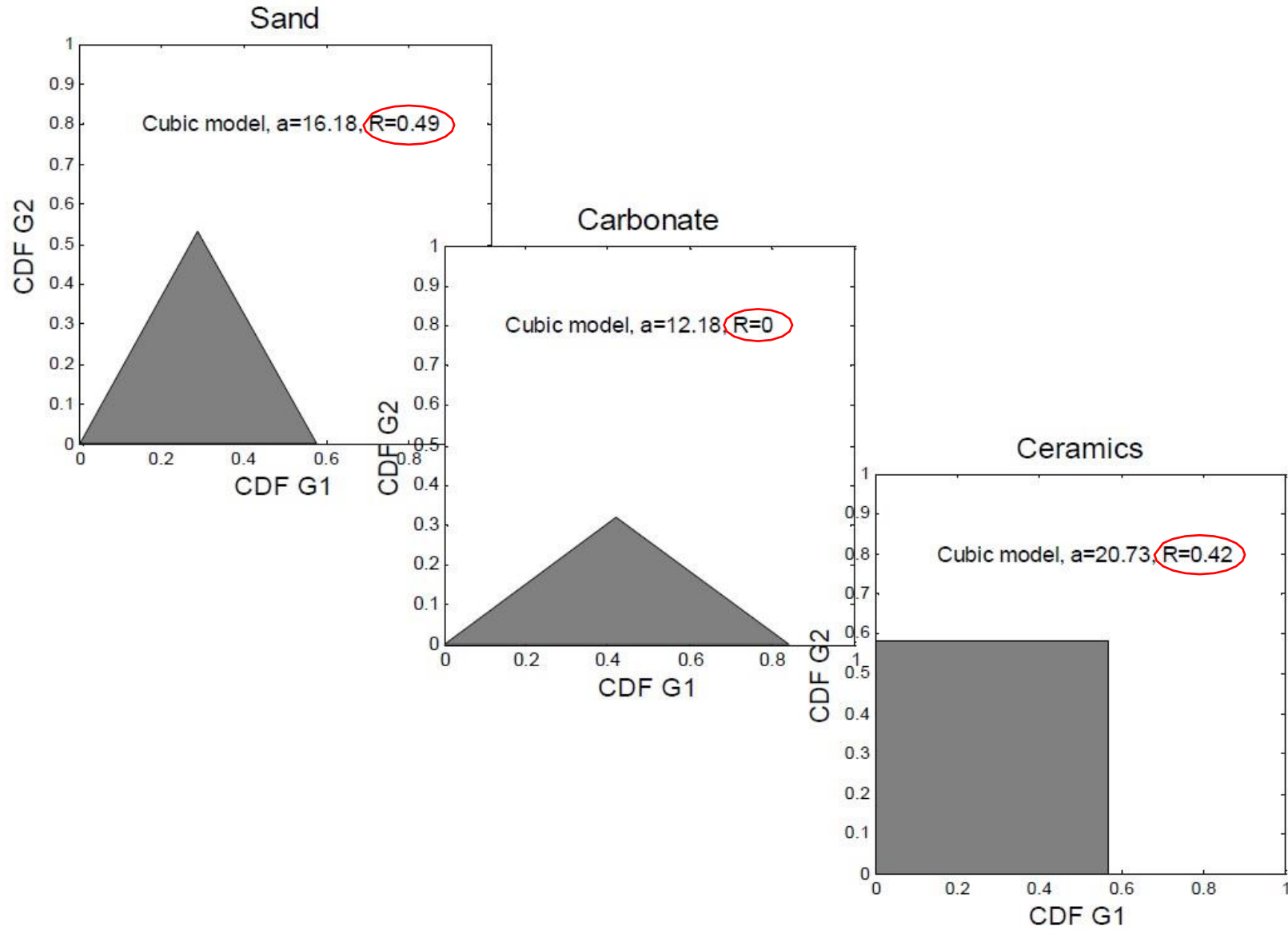
Tests : identifier l'image réelle parmi les 4 images

Image réelle : une  
céramique  
Noir : les pores



## 2.3 Simulation plurigaussienne

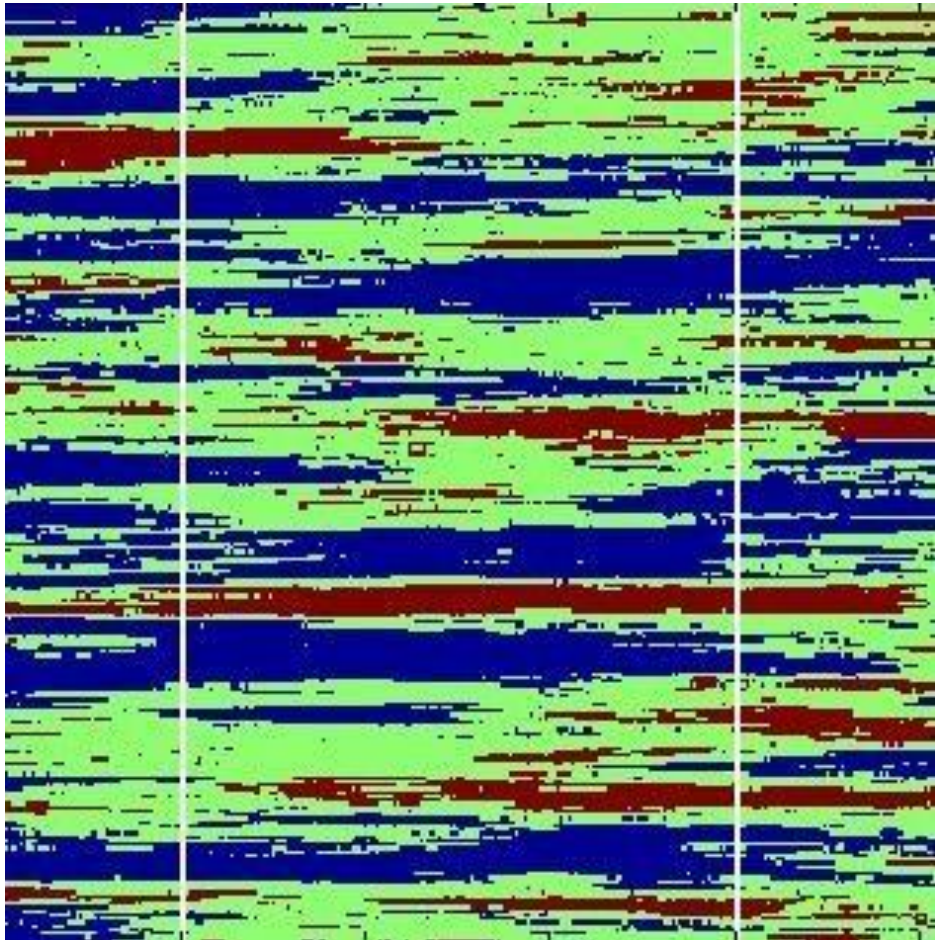
Tests : identifier l'image réelle parmi les 4 images



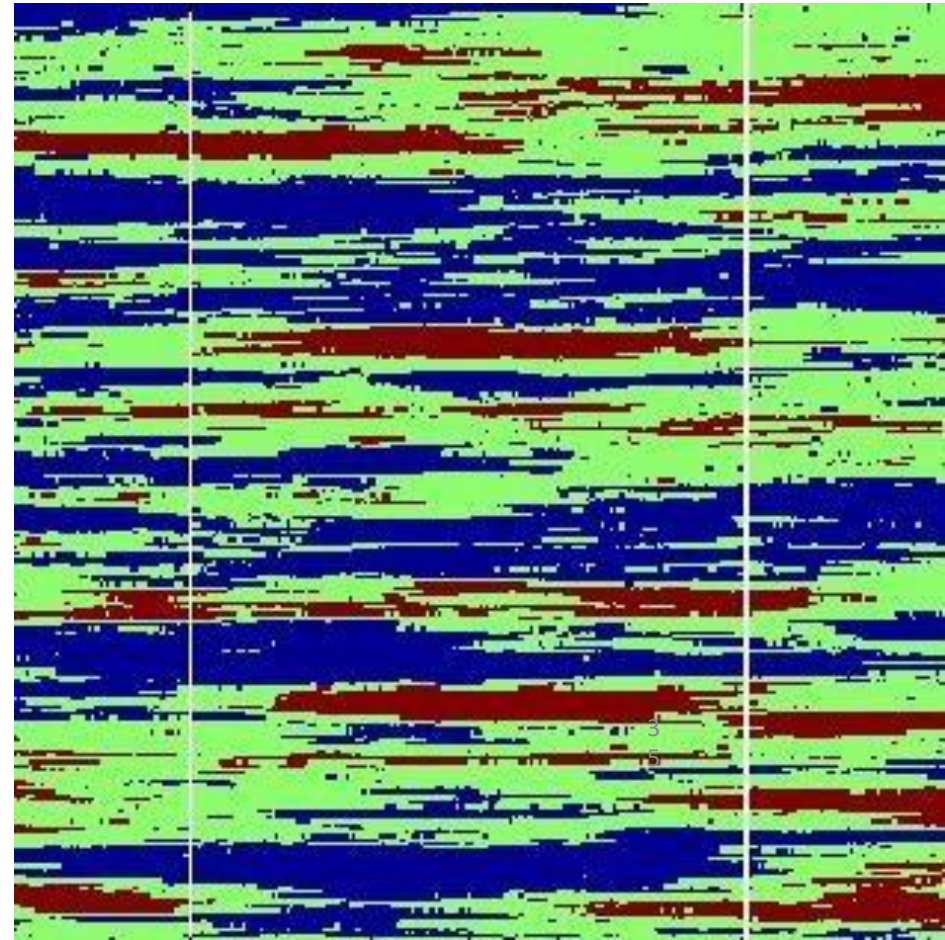
# 3. Conditionnement aux faciès observés

b) Tenir compte des faciès aux points échantillonnés :

Réalité



Faciès simulé



# 3. Conditionnement aux faciès observés

## b) Tenir compte des faciès aux points échantillonnés :

**Il existe deux méthodes pour générer des simulations de faciès conditionnelles aux faciès observés. Les deux sont itératives.**

### 1. L'échantillonneur de Gibbs

- a. Génère un champ dont les contraintes sont respectées dès la première itération;
- b. Par la suite, l'échantillonneur de Gibbs itère afin d'introduire la corrélation spatiale.

### 2. Le calage par S-STBM

- a. Génère un champ dont la corrélation spatiale est assurée par construction;
- b. Par la suite, le calage itère afin de respecter les contraintes, soit les faciès observés.

# 3.1 Échantillonneur de Gibbs

## b) Tenir compte des faciès aux points échantillonnés :

Soit  $Z(x)$  un champ gaussien, soit  $z(x_i), i = 1, \dots, N$ ,  $N$  faciès observés et soit un patron de codage  $\mathcal{U}$  :

On cherche à conditionner  $Z(x)$  aux observations  $z(x_i)$  selon le patron de codage  $\mathcal{U}$ .

Initialisation : Choisir aléatoirement des valeurs gaussiennes dans l'intervalle appropriées (selon le patron de codage  $\mathcal{U}$ ) compte tenu des faciès observés;

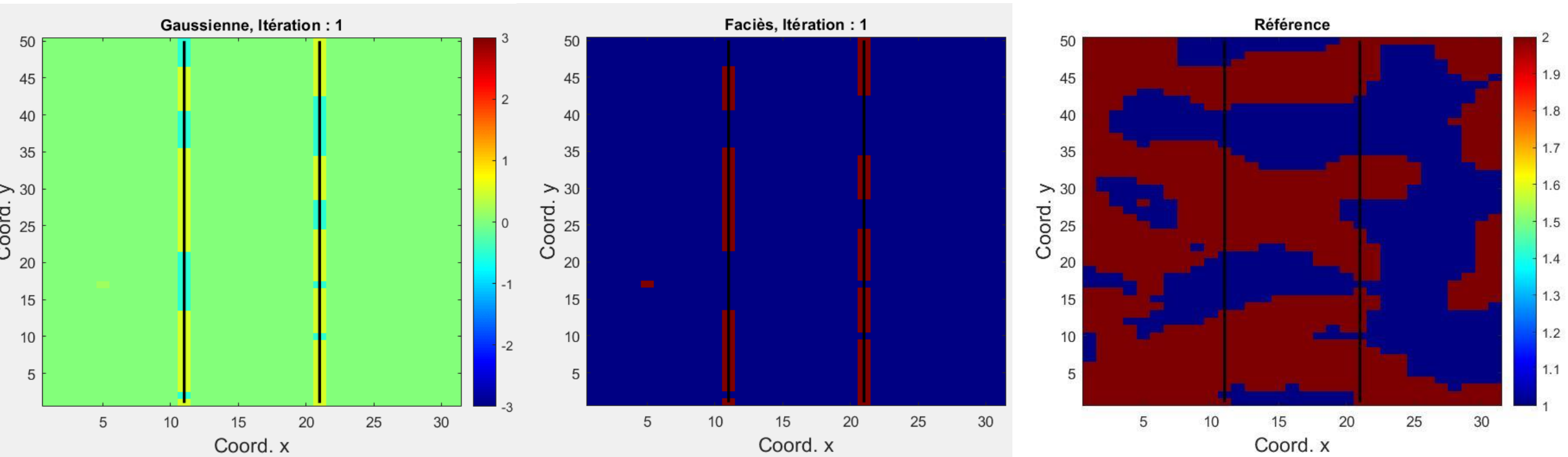
1. Sélectionner un point aléatoirement, le retirer et estimer par KS sa distribution conditionnelle;
2. Tirer une valeur aléatoire de la distribution conditionnelle en respectant les contraintes en ce point.
3. Remplacer l'ancienne valeur par celle tirée de la distribution conditionnelle ;
4. Évaluer un critère d'arrêt: arrêter ou retourner à 1)

# 3.1 Échantillonneur de Gibbs

## b) Tenir compte des faciès aux points échantillonnés :

Animation : conditionner une simulation (31x50) à deux forages par calage.

Attention les figures ne sont pas à l'échelle.



Note : l'algorithme n'a pas encore atteint la corrélation spatiale désirée. La convergence sera atteinte environ autour de 72 000 000 itérations. Lorsque chaque point aura été visité 40 000 fois. Peut-être même 100 000 visites seront requises par point. L'échantillonneur de Gibbs est généralement très lent. Ici, chaque point a été visité 100 fois seulement.

# 3.1 Échantillonneur de Gibbs

## b) Tenir compte des faciès aux points échantillonnés :

Comme pour la SGS, tirer de la distribution conditionnelle assure que l'on génère un champ gaussien ayant la bonne structure spatiale.

- Critère d'arrêt :

- Il faut **visiter un très grand nombre de fois** tous les points (voir des milliers de fois);
- Les valeurs sont **modifiées** à chaque itération d'une **toute petite quantité** (puisque la variance conditionnelle est la variance de krigeage et que les données observées sont proches);
- La **perturbation du champ est lente**. Il faut plusieurs modifications sur chaque point pour s'approcher d'un état stationnaire comme souhaité.

## 3.2 Calage par S-STBM

### b) Tenir compte des faciès aux points échantillonnés :

Il existe aujourd'hui une alternative beaucoup plus rapide que l'échantillonneur de Gibbs pour contraindre des champs à des faciès observés.

L'échantillonneur de Gibbs assure par construction l'observation des faciès et introduit itérativement la corrélation spatiale.

Une alternative serait d'assurer la corrélation spatiale par construction et de perturber les valeurs gaussiennes afin qu'ils correspondent au patron de codage désiré.





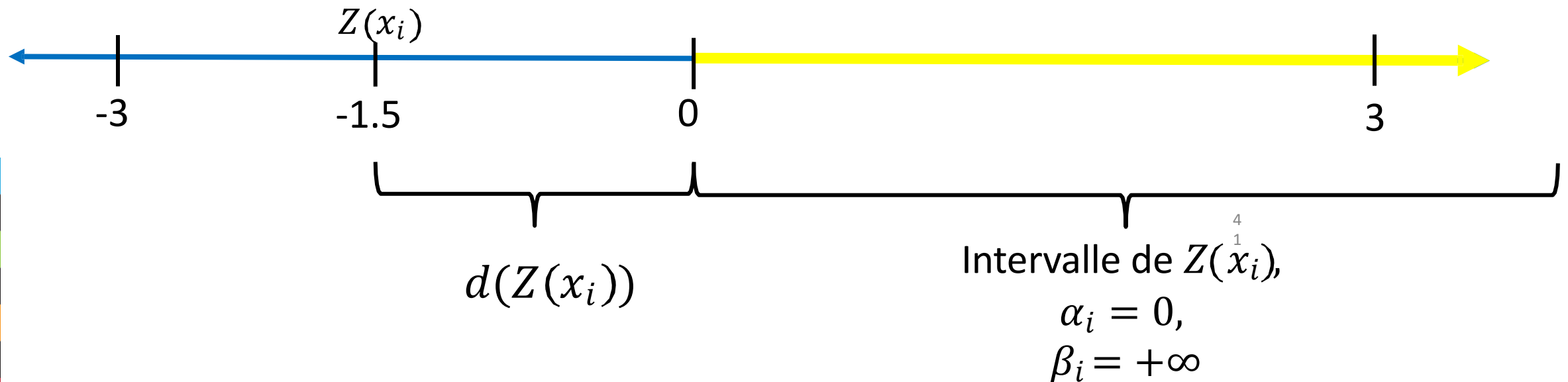
## 3.2 Calage par S-STBM

### b) Tenir compte des faciès aux points échantillonnés :

Soit le faciès jaune observé au point  $x_i$

On veut donc perturber la valeur gaussienne  $Z(x_i)$  afin de la déplacer vers l'intervalle  $y$  en jaune.

On peut mesurer la distance entre l'intervalle et  $Z(x_i)$

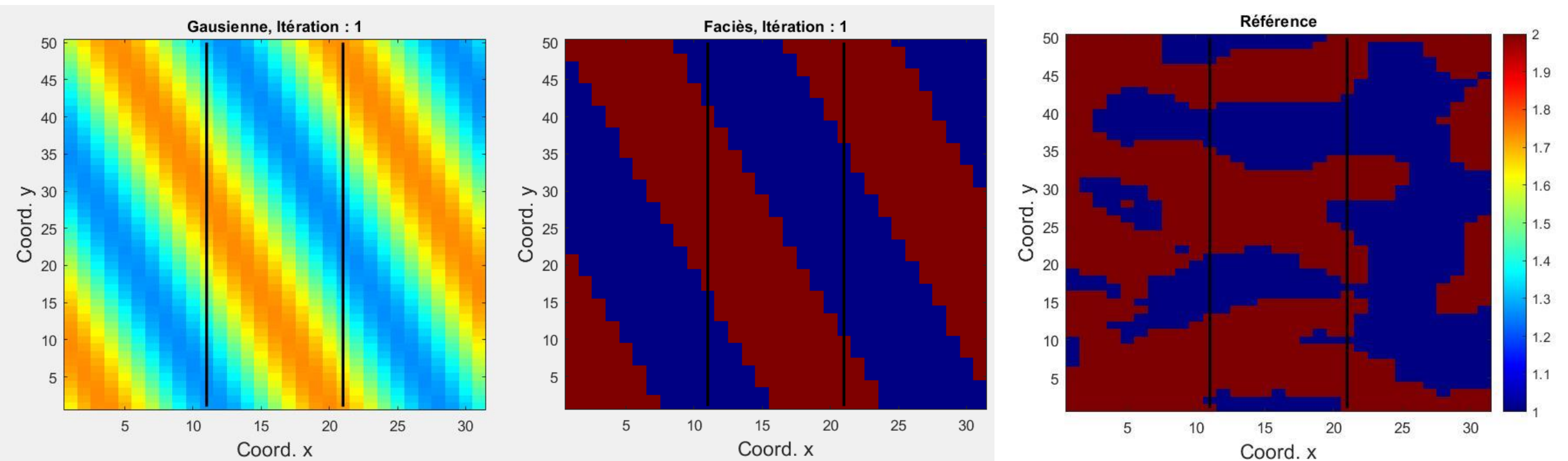


## 3.2 Calage par S-STBM

### b) Tenir compte des faciès aux points échantillonnés :

Animation : conditionner une simulation (31x50) à deux forages par calage.

Attention, les figures ne sont pas à l'échelle.



Note : on aurait pu arrêter l'algorithme après 1 000 itérations. Contrairement au Gibbs, le calage par S-STBM perturbe tous les points à chaque itération. Donc, on a performé ,ici, environ 1 550 000 perturbations, contrairement aux 72 000 000 requis par l'échantillonneur de Gibbs. Un gain très important. Selon la corrélation spatiale, le gain est d'environ de 2 à 3 ordres de grandeur au niveau du temps de calcul

## 3.2 Calage par S-STBM

### b) Tenir compte des faciès aux points échantillonnés :

Cette alternative est très rapide lorsque combinée avec la méthode des bandes tournantes spectrales séquentielles (S-STBM, Lauzon et Marcotte (2020)).

Il suffit de minimiser la fonction objectif suivante :

$$FO(Z(x)) = \lambda \underbrace{\left[ \frac{1}{N} \sum_{i=1}^N (\max\{\alpha_i - Z(x_i), 0\} + \max\{Z(x_i) - \beta_i, 0\}) \right]}_{\text{Terme pour respecter les faciès observés}} + (1 - \lambda) \underbrace{|s_j^2 - s_{j,u}^2|}_{\text{Terme pour préserver la variance de la simulation}}$$

$\alpha_i$  et  $\beta_i$  : Borne inférieure et supérieure de l'intervalle au point  $x_i$ .

## b) Tenir compte des faciès aux points échantillonnés :

Quoi retenir :

### L'échantillonneur de Gibbs :

- Exact;
- Simple à programmer;
- Séquentielle, point par point;
- Très lent à introduire la corrélation spatiale.

### Calage

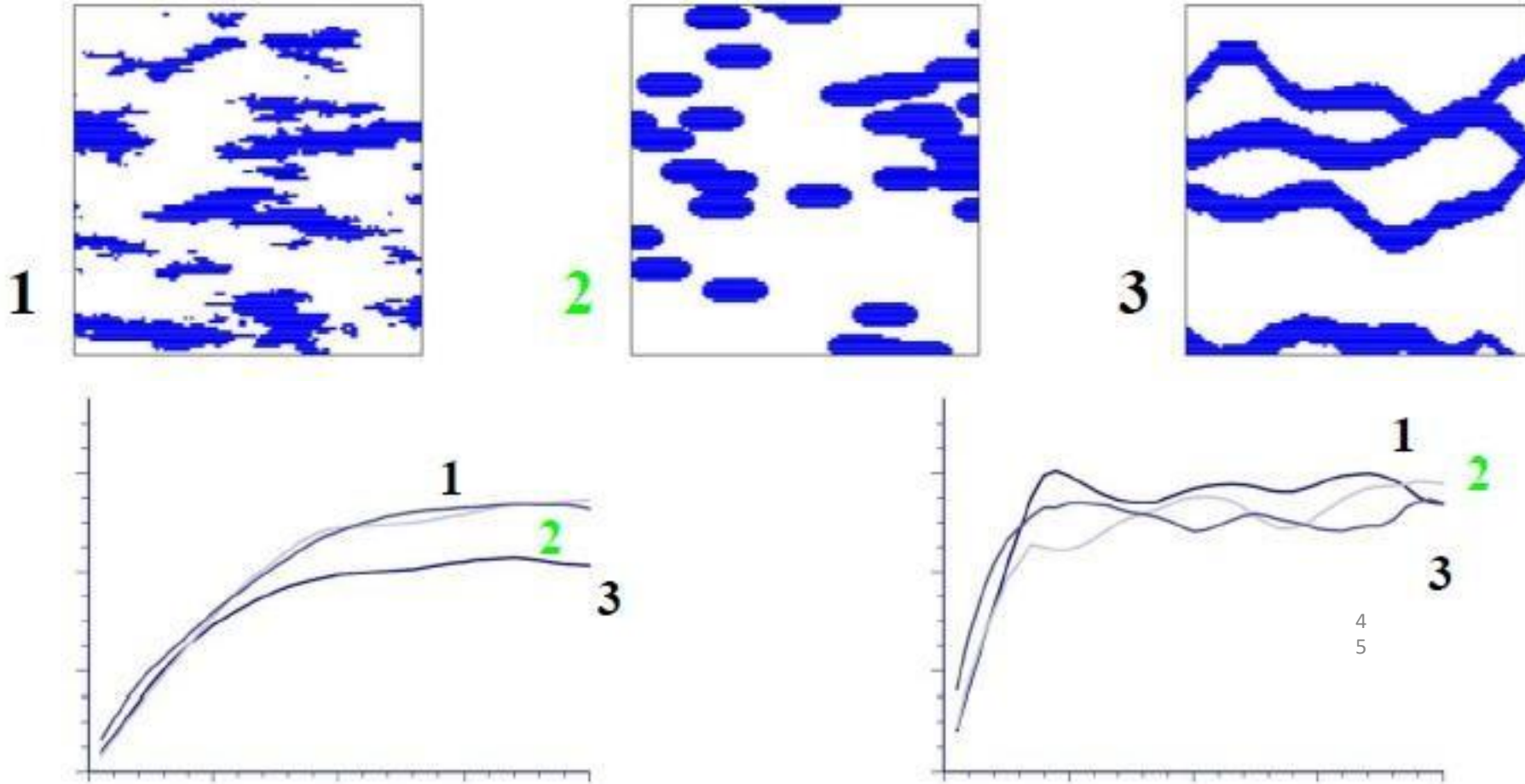
- Approximatif (recours au T-SGS pour rendre exact);
- Complexe à programmer;
- Les perturbations sont globales lorsqu'appliquées avec S-STBM;
- Très rapide comparativement à l'échantillonneur de Gibbs.



# 4. Méthode multipoints

Problème :

Le variogramme ne suffit pas à saisir toutes les caractéristiques d'un champ donné

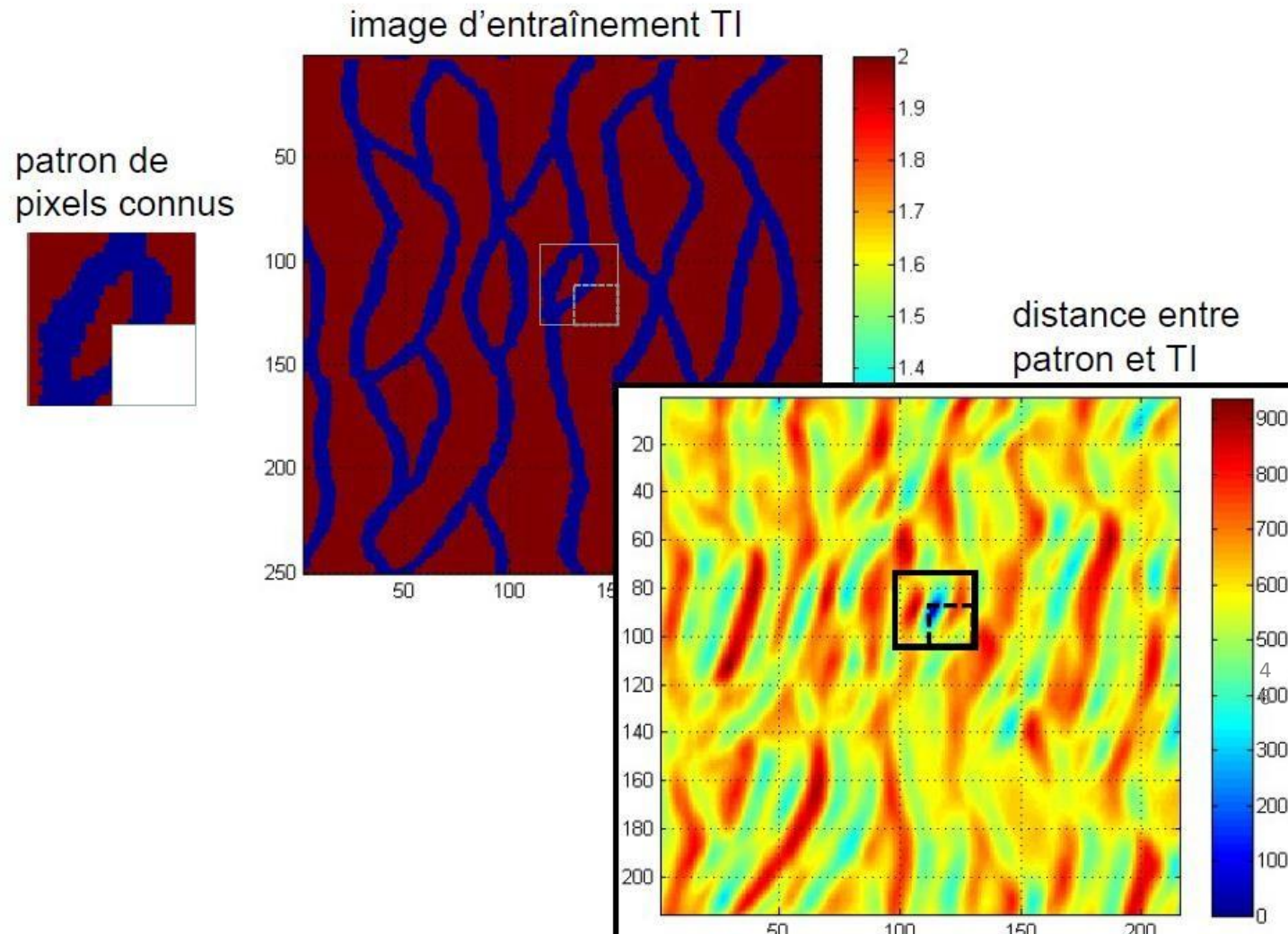


D'après Caers and Zhang, (2004)

# 4. Méthode multipoints

Idée :

Utiliser des images d'entraînement qui servent de base pour générer des configurations possibles de faciès



# 4.1 SNESIM

L'idée originale est de Guardiano et Srivastava (1992). Soit :

- Un champ de pixel soit noir (1) ou blanc (0) dont on veut simuler le pixel  $k$ ;
- $S_n$  l'état correspondant à un ensemble de  $n$  pixels;
- $S_k$  l'état du pixel  $k$ ;
- $S_{k,n}$  l'état correspondant au même ensemble de  $n$  pixels plus le pixel  $k$ .

On a :

$$S_{k,n} = S_k \cap S_n$$

Utilisant la définition de probabilité conditionnelle, on a :

$$P(S_{k,n}|S_n) = \frac{P(S_k \cap S_n)}{P(S_n)}$$



Pour estimer ces probabilités, on utilise des modèles analogues (Images d'entraînement).

# 4.1 SNESIM

## Algorithme :

On effectue une visite aléatoire de tous les points à simuler  $x_i$  ;

- i. On définit un voisinage au point  $x_i$  . On identifie dans ce voisinage tous les points  $x_j$  ,  $j \neq i$  , déjà simulés ou connus et on note leurs états. Cela définit  $S_n$ ;
- ii. On cherche dans le modèle analogue toutes les occurrences de  $S_n$ . Pour chaque occurrence, on note la valeur de  $S_k$ ;
- iii. On tire aléatoirement selon  $P(S_{k,n}|S_n) = \frac{P(S_k \cap S_n)}{P(S_n)}$ ;
- iv. On revient à i jusqu'à ce que l'on ait visité tous les points à simuler.



# 4.1 SNESIM

## Exemple :

Soit l'image d'entraînement suivante :

0 0 1 1 0 1 1 0 0 0 0 1 1 1 1 0 1 0 1 0 0 0 0 1 1 0 1 1 1 0 0 1 1 1 0 0 0 0 0 1 1

On a observé la séquence 0 1  $x_0$  0 1 (i. e.,  $S_n$ ). Quel faciès simuler au point  $x_0$ ?

- 1) On parcourt la TI à la recherche de signal présentant la configuration 0 1 ? 0 1;
- 2) On identifie les occurrences  $S_{k,n}$ ;
- 3) On simule le faciès 1 au point  $x_0$ , car seulement le faciès 1 est observé pour la séquence 0 1 ? 0 1.

# 4.1 SNESIM

## Problème d'occurrence: Délestage

Très souvent on ne trouve aucune occurrence du patron recherché dans la TI.

Deux strategies :

- Réduire le voisinage en enlevant les points les plus éloignés (SNESIM)
- Calculer une distance entre l'état  $S_{k,n}$  de la simulation et l'état  $S_n$ , sur les mêmes pixels, dans la TI (FilterSIM)

Dans ce cas on prend le groupe de pixels étant le plus proche ou on en choisit un au hasard parmi les plus proches.

Ces deux stratégies font que l'on reproduit moins bien la structure multipoints.

On n'a pas un bon contrôle sur la variabilité des réalisations obtenues. Souvent les réalisations montrent de grandes similarités l'une envers l'autre, ce qui est contraire à l'objectif poursuivi.

# 4.1 SNESIM

## Problème d'occurrence: Structure informatique

Très inefficace de parcourir l'image à nouveau à chaque fois que l'on doit simuler un point.

**Scanner la TI une seule fois** et enregistrer dans une structure informatique (typiquement, un arbre ou une liste) l'ensemble des états dont on va avoir besoin en cours de simulation.

Si la structure est bien conçue, on pourra efficacement la parcourir pour retrouver les fréquences recherchées. Par exemple Strebelle (2001) décrit une structure en arbre.



# 4.1 SNESIM

## Problème d'occurrence: Multigrilles

**Problème:** Ignorer les points éloignés affecte la reproduction des structures à grande échelle.

**Solution:** Utilisation de multigrilles imbriquées pour simuler à différentes résolutions.

**Étapes de Simulation:** Simuler 1 pixel tous les 16. Puis, 1 pixel tous les 8. Ensuite, 1 pixel tous les 4. Continuer avec 1 pixel tous les 2. Enfin, simuler toute la grille.

**Avantages et Inconvénients:** Améliore la reproduction des structures à grande échelle. Peut sacrifier la précision des textures à petite échelle.

**Exigences:** Chaque grille intermédiaire nécessite un arbre de recherche spécifique et l'application de l'algorithme de base à chaque étape.



## 4.2 Approche séquentielle par paquet (FilterSim)

### Idée :

Pour mieux **reproduire la texture petite échelle**, il peut être intéressant de simuler les pixels par paquets (groupes de pixels contigus).

Dans ce cas, on **définit une distance** entre l'état de la simulation et les états du modèle analogue aux mêmes pixels (sur  $S_n$ ).

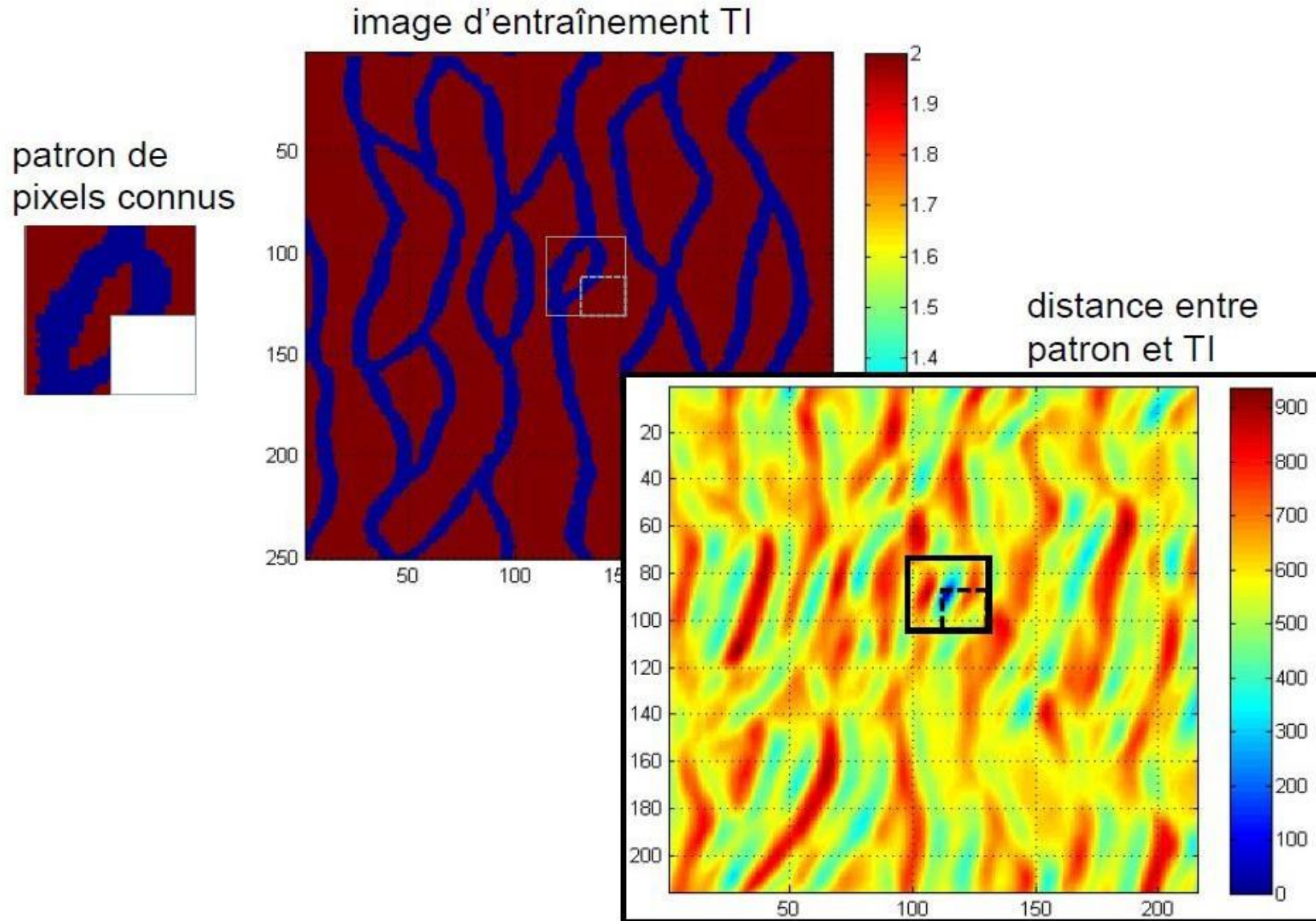
Comme les paquets sont collés ensembles dans la simulation, on est **assuré d'avoir la bonne texture à l'intérieur du paquet simulé**.

**Cependant, entre deux paquets simulés rien ne garantit que la jonction soit réaliste.**



# 4.2 Approche séquentielle par paquet (FilterSim)

Exemples d'occurrences d'un bloc :



## 4.2 Approche séquentielle par paquet (FilterSim)

### Algorithme :

#### Initialisation

- on définit des mesures synthèses (filtres) sur chaque paquet (moyenne, dérivées directionnelles, courbures,...); dans le cas continu, si l'on a L filtres, on obtient L valeurs pour chaque paquet;
- on fait une classification (K-means) basée sur ces mesures pour obtenir des groupes de paquets semblables ;
- on effectue une moyenne dans chaque groupe de paquets pour obtenir un paquet prototype (ou archétype) par groupe;

#### Définition :

- Un pixel déjà simulé est de type  $t_1$  (données observée « hard »),  $t_2$  (donnée simulée « hard ») ou  $t_3$  (donnée simulée « soft »).
- Un pixel simulé « hard » est un pixel faisant partie de la zone interne d'un paquet.
- Un pixel simulé « soft » fait partie de la couronne externe du même paquet.



## 4.2 Approche séquentielle par paquet (FilterSim)

### Algorithme :

#### Simulation

- On visite un pixel «  $u$  » choisi au hasard (parmi tous les pixels qui ne sont pas  $t_1$  ou  $t_2$ );
- On identifie les pixels informés (simulés ou connus) dans le paquet centré sur  $u$  ;
- On calcule la distance entre les pixels trouvés dans le voisinage de  $u$  et les mêmes pixels de chaque prototype. Dans le calcul de cette distance, on donne un poids plus grand aux pixels  $t_1$  puis  $t_2$  puis  $t_3$  ;

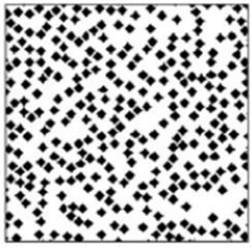
Soit  $p_{min}$  le prototype dont la distance aux pixels du voisinage est la plus petite:

- On pige au hasard un paquet dans le groupe dont le prototype est le plus près du paquet simulé;
- On colle le paquet en entier sauf pour les pixels déjà identifiés  $t_1$  et  $t_2$  ; la partie centrale est considérée comme  $t_2$  (et ne sera plus modifiée), la partie externe comme  $t_3$  (et donc non simulée encore).

Note : Arpat et Caers (2007) proposent de directement calculer les distances sur les paquets de pixels et de choisir parmi les paquets dont la distance est inférieure à un seuil. Bref, ils évacuent l'idée des filtres et de la classification préliminaire dans filtersim. (Arpat, Caers, 2007. Conditional simulation with patterns. Math. Geol. 39, 2, 177-204.)

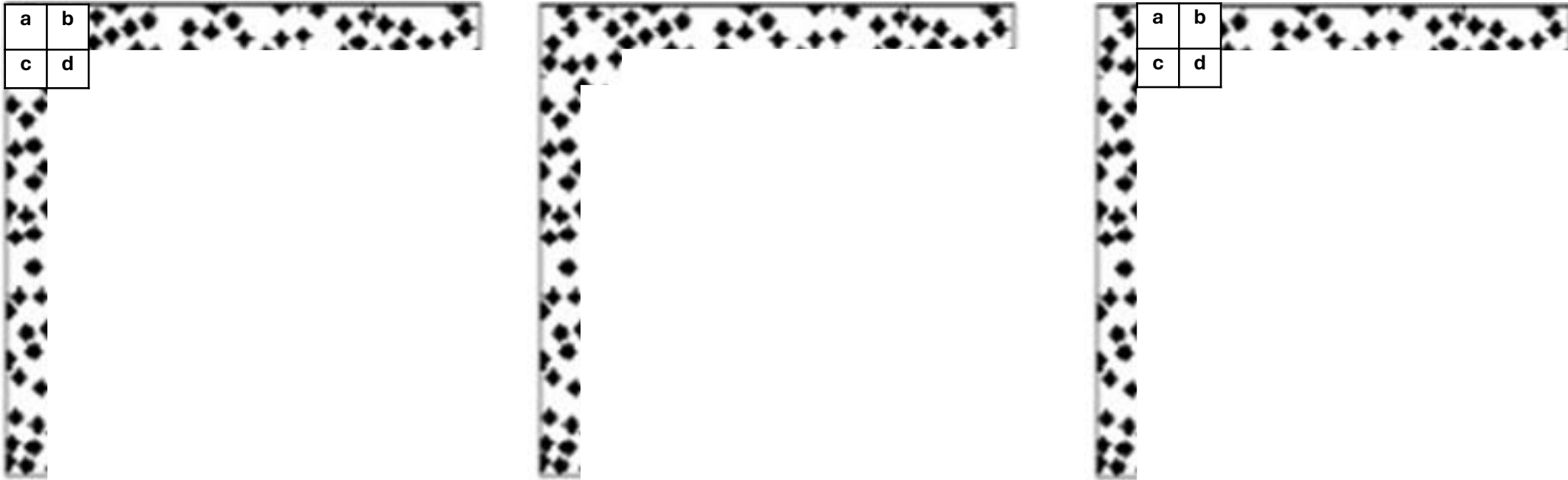


## 4.2 Approche séquentielle par paquet (Patchwork)

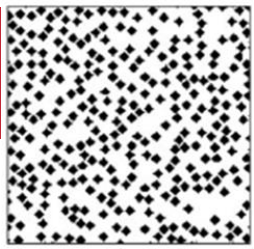


### Algorithme :

L'idée de l'algorithme est de **procéder de façon systématique**, plutôt qu'aléatoire, du coin supérieur gauche vers le coin inférieur droit. On utilise un patron carré, chaque zone a, b, c, d dans le dessin suivant représentant plusieurs pixels (e.g. 6 x 6 pixels, 8 x 8 pixels, 16 x 16 pixels) :



# 4.2 Approche séquentielle par paquet (Patchwork)



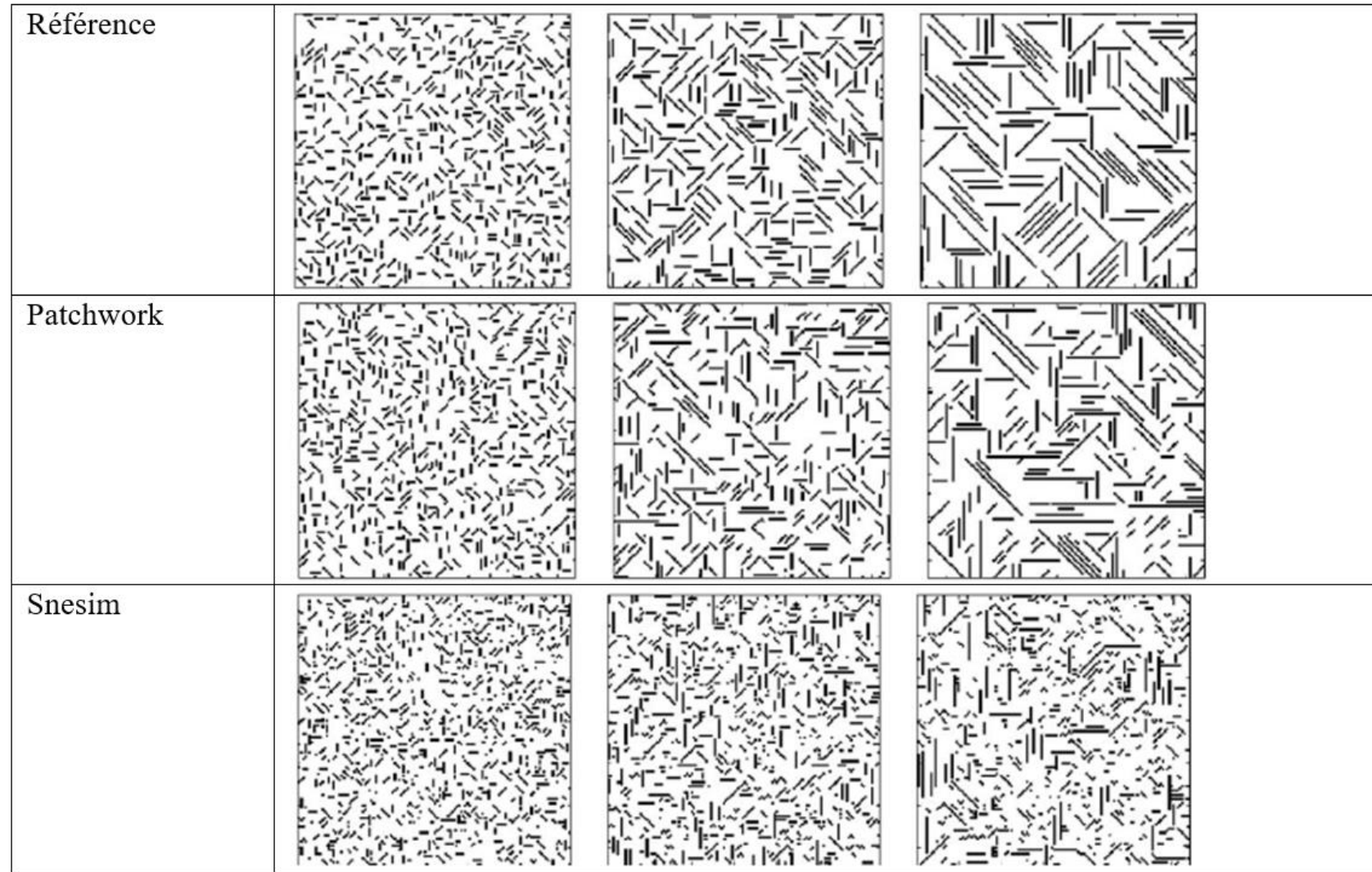
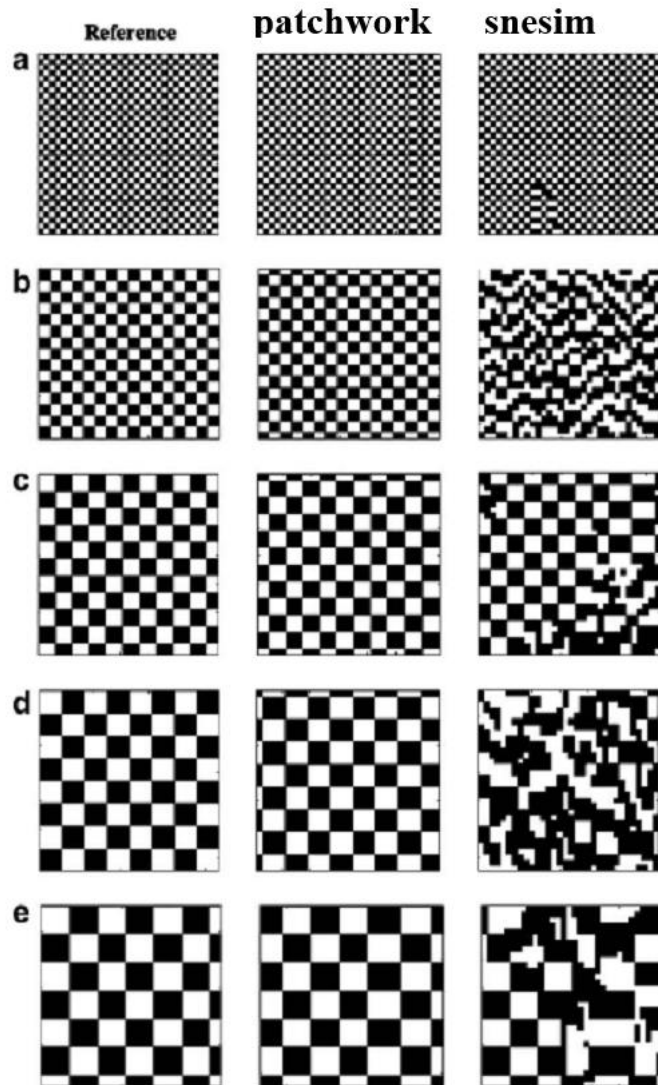
Algorithme :

|           |  |  |  |
|-----------|--|--|--|
| Référence |  |  |  |
| Patchwork |  |  |  |
| Snesim    |  |  |  |



# 4.2 Approche séquentielle par paquet (Patchwork)

Algorithme :



# 4.2 Approche séquentielle par paquet (Patchwork)

## Généralisation :

### Simulation de variables continues

- Généralisation directe avec les méthodes basées sur des distances

### Extension au 3D

- Tous les modèles, mais la généralisation se fait plus facilement avec SNESIM
- La principale difficulté est d'avoir une image d'entraînement. Des méthodes utilisent des TI 2D pour simuler des modèles 3D.

### Données « soft »

- Plus immédiate avec FilterSIM

### Non-stationnarité

- Complexe, mais possible avec SNESIM (Boucher, 2009). Une variable indicatrice est simulée en premier lieu indiquant quel arbre ou quelle liste utiliser pour la simulation

### Contrôle des moments d'ordres inférieurs

- Ils ont tendances à se détérioré. À noter que lorsque l'on essaye de bien contrôler les statistiques d'ordre inférieur, plus on perd sur la qualité de la reproduction des textures et structures.



## 4.2 Approche séquentielle par paquet (Patchwork)

### Généralisation :

Simulation

➤ Gé

Extension

➤ To

➤ La

sim

Données «

➤ Plu

Non-statio

➤ Co

pre

Contrôle d

➤ Ils

d'o

***Le direct sampling***  
**permet de gérer tous**  
**ces problèmes.**  
**(prochain cours)**

D pour

lée en

stiques



# 4. Méthode multipoints

## Approches : deux grandes classes

### 1) Pixel par pixel (p. ex. *direct sampling* de G. Mariethoz )

#### Avantages

- Conditionnement assuré
- Plus adaptable localement
- Extension au cas 3D plus directe

#### Inconvénients

- Plus lent
- Présence de discontinuités visibles sur les images simulées

### 2) Groupe de pixels contigus ( bloc ou pièce )

#### Avantages

- Plus rapide
- Images simulées plus continues
- Permet de générer des images alternatives à la TI

#### Inconvénients

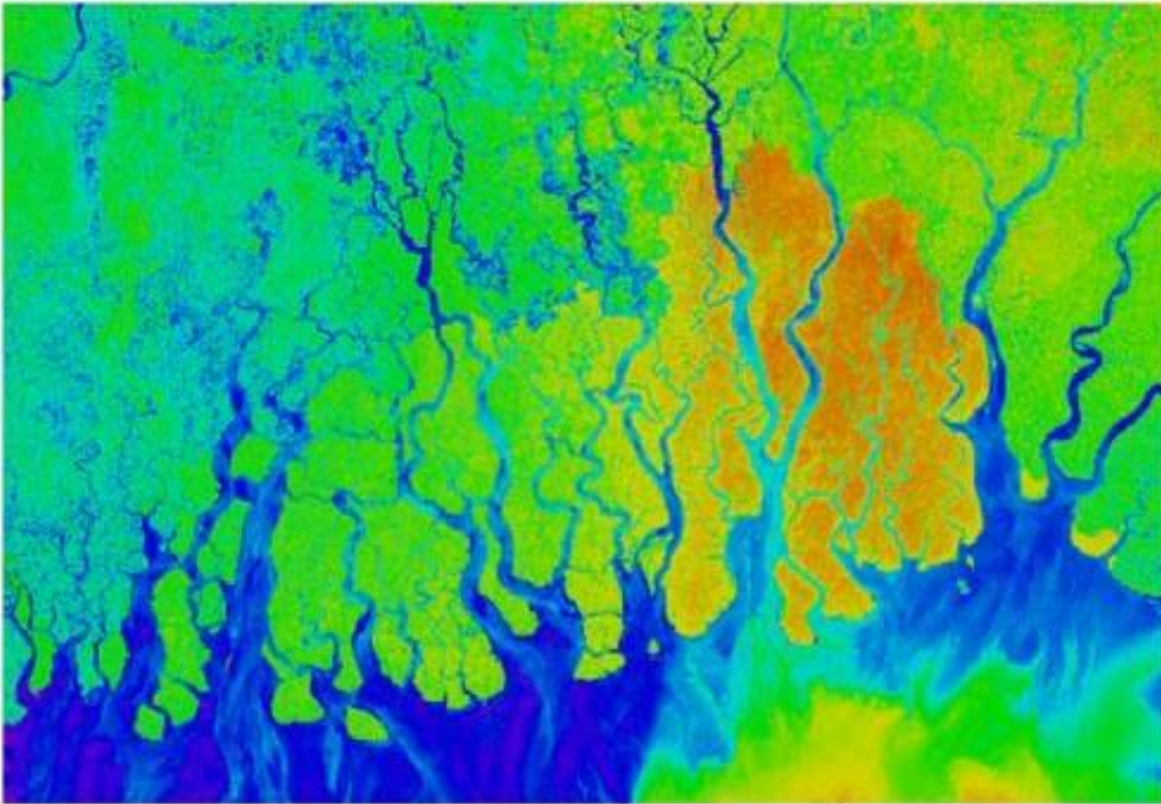
- Conditionnement plus difficile
- Plus rigide
- Extension au cas 3D possible, mais plus complexe



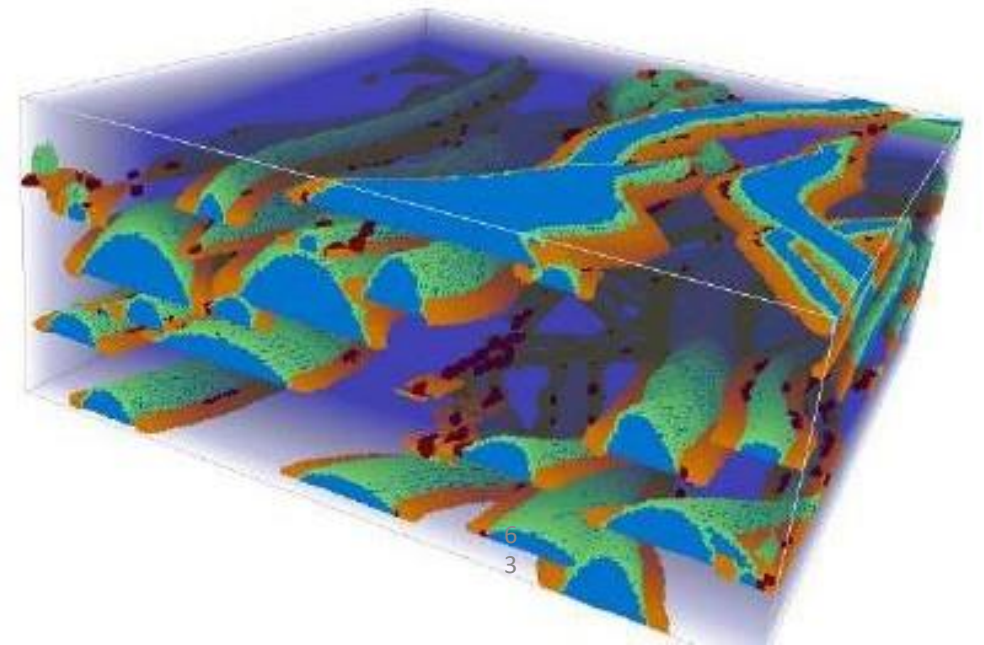
# 4. Méthode multipoints

Sources possibles de TI :

En 2D souvent : images satellites, images d'affleurement



en 3D: des modèles d'objets simulés



# 5. Démo - ArchPy

Motivation : Automatiser la génération de modèle géologique à partir d'une base de données de forages

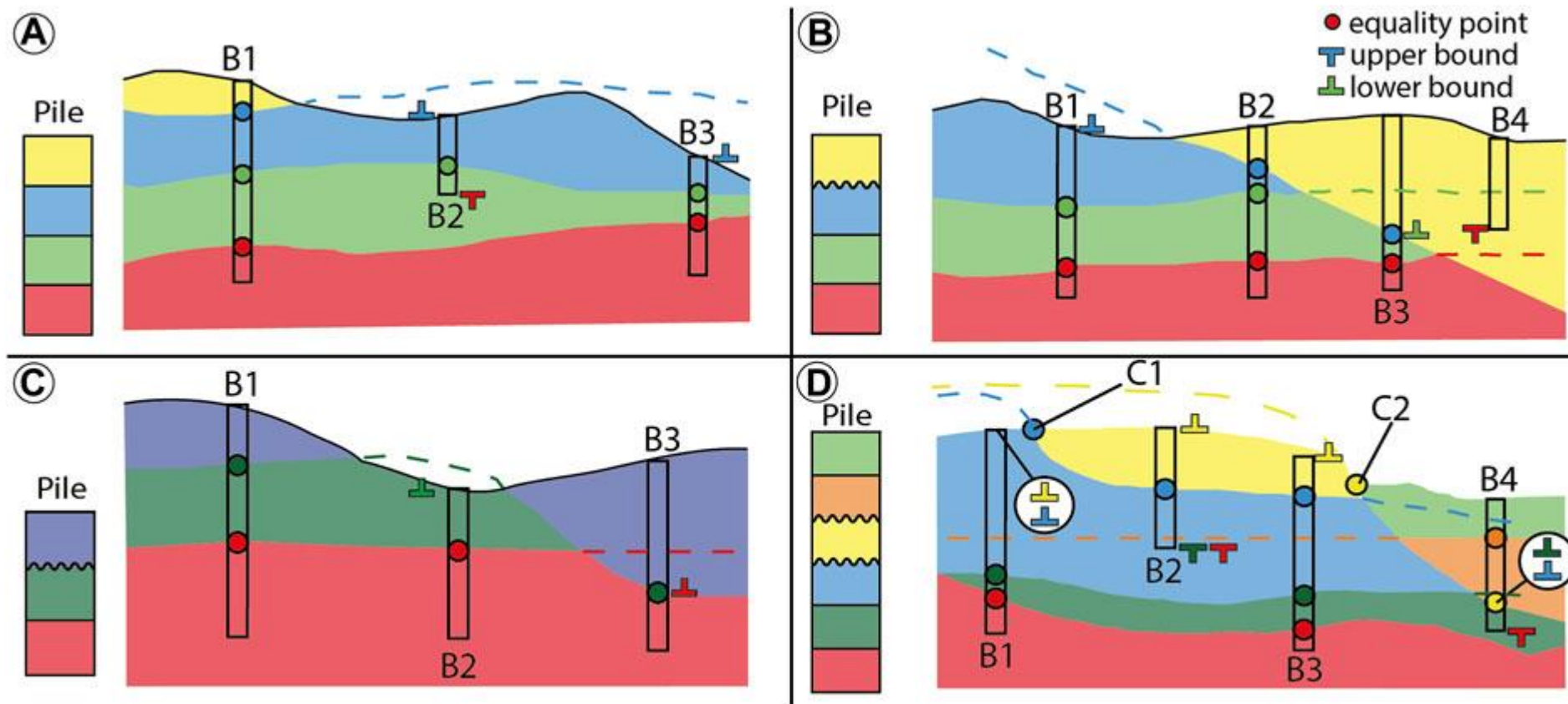
Une simulation complète se déroule hiérarchiquement en trois étapes principales :

- 1) Simuler les **surfaces** délimitant les limites des unités et permettant ainsi de définir le domaine stratigraphique de l'unité ;
- 2) Simuler les **faciès** de remplissage de chaque unité en utilisant différentes méthodes géostatistiques en fonction des connaissances géologiques préalables ;
- 3) Simuler les **propriétés** à l'intérieur de chaque faciès indépendamment.



# 5. Démo - ArchPy

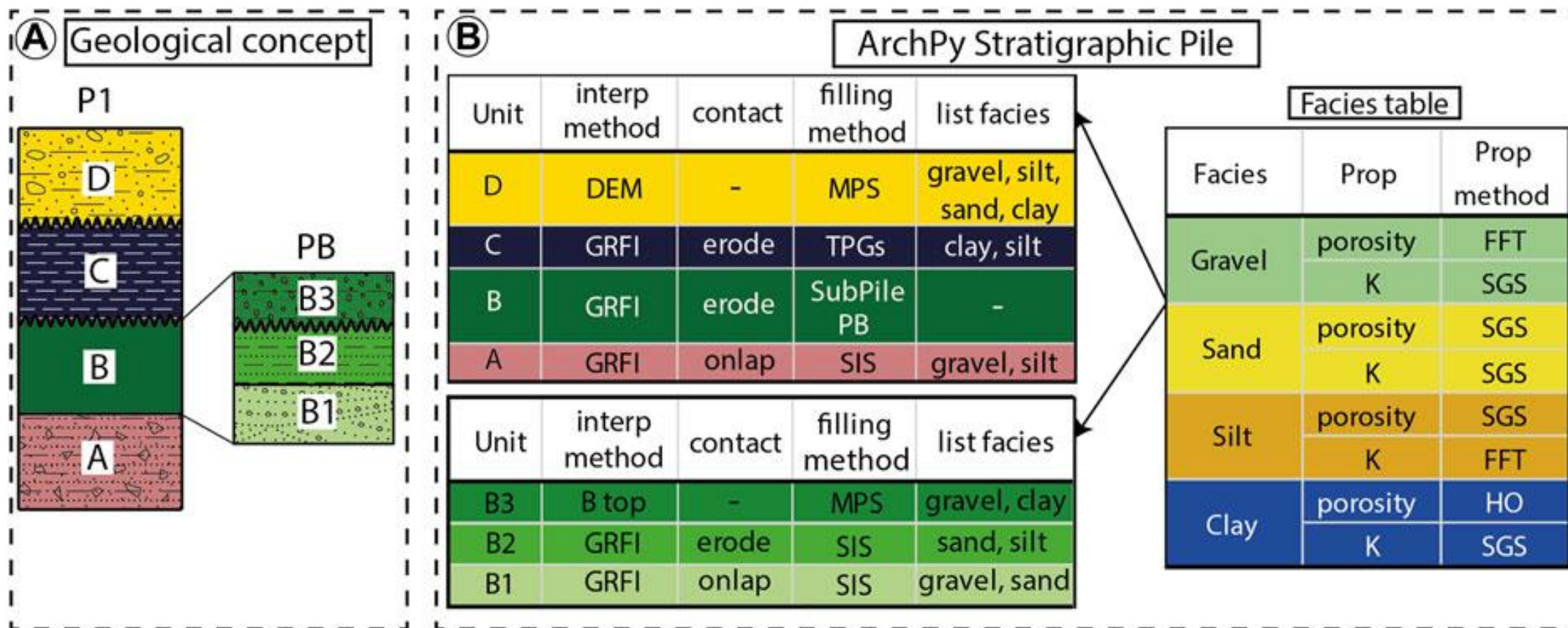
## Données de conditionnement : Gibbs



**FIGURE 5.** Four examples (A–D) of how inequalities and equalities are extracted from boreholes and field data. (B) indicates borehole information and (C) a unit contact (e.g., observed on the field). For each example, a stratigraphic pile is defined to indicate the relationships between the units and the nature of the surface contact (straight line is onlap and corrugated is erode). Dashed lines represent simulated surfaces before applying erosion rules (see Section 2.3).

# 5. Démo - ArchPy

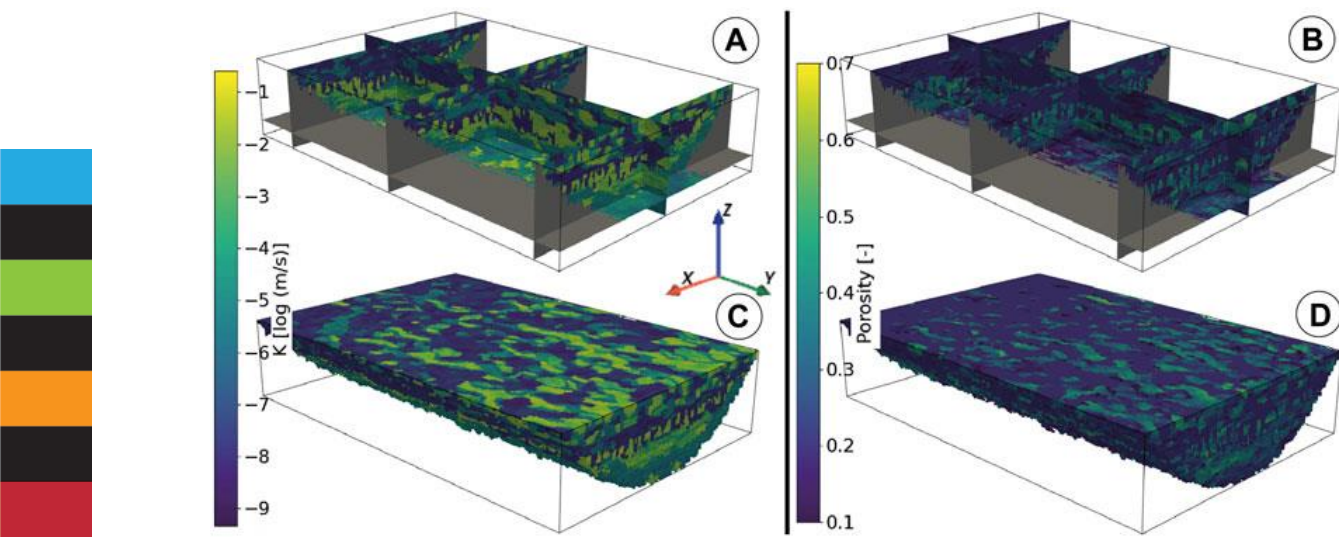
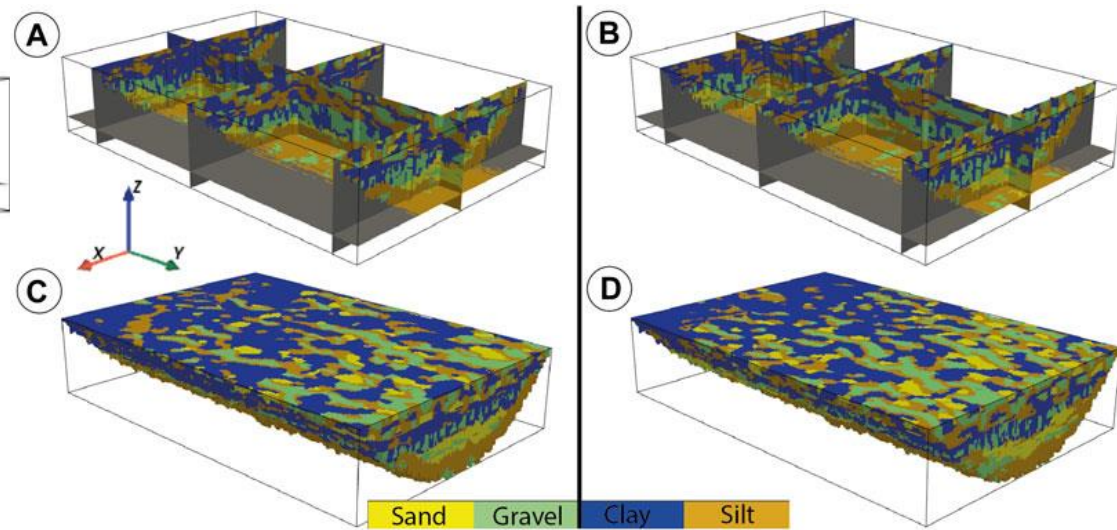
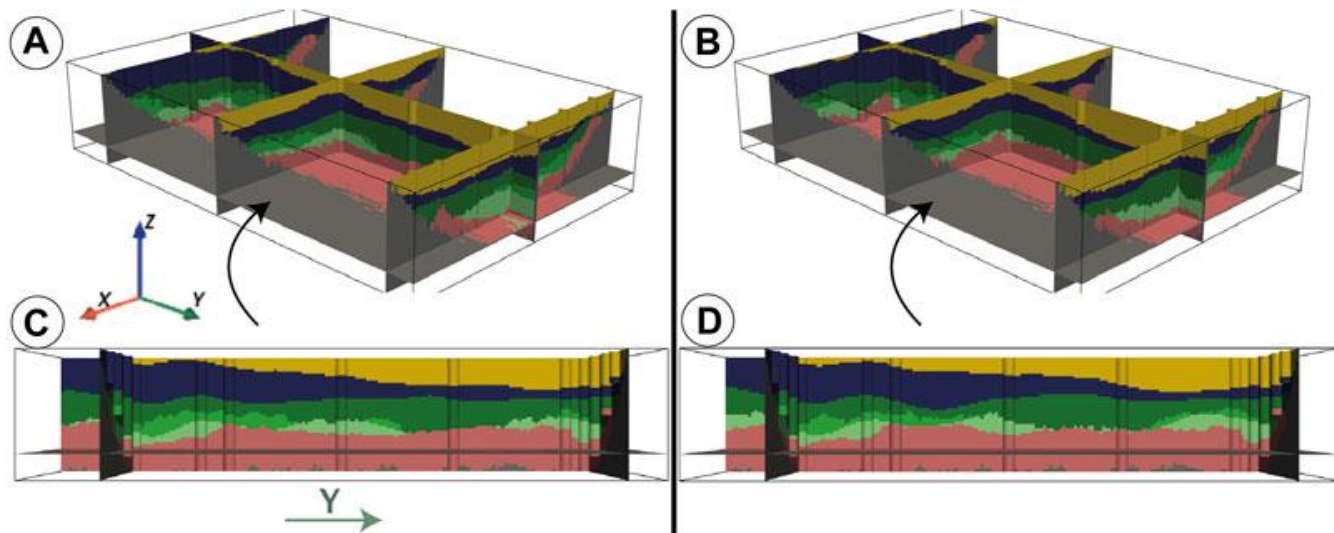
## Pile Stratigraphique :



**FIGURE 2.** Schematic representation of an ArchPy stratigraphic pile **(B)** given a geological concept **(A)**; interp. method, interpolation method; GRF, Gaussian random functions; DEM, digital elevation model; Prop method, property simulation method; MPS, multiple-points statistics; SIS, sequential indicator simulation; TPG, truncated pluri-Gaussians; HO, homogeneous; K, hydraulic conductivity; FFT, fast Fourier transform; SGS, sequential Gaussian simulation.

# 5. Démo - ArchPy

## Pile Stratigraphique :



# 5. Démo - ArchPy

## Pile Stratigraphique :

