Méthodes de simulation multipoints

Motivation : Les méthodes habituelles permettent de reproduire les statistiques d'ordre 2 (histogramme et covariance ou variogramme) mettant en relation deux points à la fois. Or, plusieurs images très différentes par leur structure et leur texture peuvent montrer les mêmes histogramme et variogramme. Des propriétés comme la connectivité des zones conductrices d'un aquifère dépendent de statistiques mettant en jeu plusieurs points. On veut donc reproduire ce genre de statistiques.

Préalable : On devra disposer d'informations pour déterminer les statistiques multipoints que l'on veut reproduire. Cette information va généralement provenir de *modèles analogues.*

Un modèle analogue est un champ sensé présenter une texture et une structure semblable à la réalité inconnue. On peut l'obtenir à partir d'affleurements, de modèles géologiques, de dessins, de simulations d'objets ou autres.

On distingue deux grandes catégories de méthodes : l'approche séquentielle ponctuelle et l'approche séquentielle par patrons. À l'intérieur de chaque catégorie il existe plusieurs variantes que nous ne détaillerons pas ici.

Approche séquentielle ponctuelle (snesim)

L'idée originale est de Guardiano et Srivastava (1992)[1]. Considérons le cas où chaque pixel est soit noir ou blanc. Soit le pixel k que l'on veut simuler. Soit S_n l'état correspondant à un ensemble de n pixels. Soit S_k l'état du pixel k et $S_{k,n}$ l'état correspondant au même ensemble de n pixels plus le pixel k. On a $S_{k,n} = S_k \cap S_n$. Un état est un ensemble de valeurs prises par les pixels considérés. Ainsi, avec n=2, 4 états sont possibles (0,0), (0,1), (1,0), (1,1) (où 1 désigne blanc et 0 noir). Pour un ensemble de n pixels, 2^n états différents sont possibles.

Utilisant la définition de probabilité conditionnelle, on a :

$$P(S_{k,n} \mid S_n) = \frac{P(S_k \cap S_n)}{P(S_n)} \qquad (1)$$

Pour estimer ces probabilités, on utilise des modèles analogues. On compte simplement la fréquence où l'on observe $S_{k,n}$ et S_n pour une configuration identique que l'on fait glisser sur le modèle analogue (note : $P(S_n) \equiv P((S_k = 0 \cap S_n) \cup (S_k = 1 \cap S_n))$

Algorithme :

i. on effectue une visite aléatoire de tous les points à simuler;
ii. à chaque point, on définit un voisinage. On identifie dans ce voisinage tous les points déjà simulés ou connus et on note leurs états. Ensemble, ces états définissent S_n;
iii. on cherche dans le modèle analogue toutes les occurrences de S_n. Pour chacune, on note aussi la valeur de S_k;
iv. on tire aléatoirement selon l'équation (1);
v. on retourne à ii. jusqu'à ce que l'on ait terminé.

[1] Guardiano, F. et Srivastava M., 1992. Multivariate geostatistics : beyound bivariate moments. Troia'92, p. 133-144, Kluwer.

Remarques :

- **Délestage.** En iii. il arrive souvent que l'on ne trouve aucun état S_n dans l'image de référence. Pour s'en convaincre, imaginons une image analogue de 256 x 256 pixels. Supposons un voisinage de 25 pixels sur une grille carrée. On pourra trouver 252 x 252 = 63504 groupes de pixels différents de 5 x 5 sur l'image analogue. Or, on peut rencontrer théoriquement jusqu'à 2^{25}>33 millions ensembles différents de valeurs, soit un ratio (états simulés possibles/états du modèle analogue) de 1.9/1000 ! On ne peut évidemment évaluer des probabilités si l'événement recherché n'est pas observé un certain nombre de fois. Lorsque l'on ne retrouve pas l'état recherché dans l'image, *deux stratégies peuvent être utilisées*. La plus simple consiste à réduire la taille du voisinage, par exemple en enlevant les pixels les plus éloignés du point que l'on veut simuler (*délestage*; approche suivie dans **snesim**). Une autre approche est de calculer une mesure de *distance* entre l'état $S_{k,n}$ de la simulation et les états S, sur les mêmes pixels, dans le modèle analogue. On choisit alors parmi les états les plus près (ou suffisamment près) de l'état de la simulation (utilisé dans **filtersim**). Ces deux stratégies font que l'on reproduit moins bien la structure multipoints.

- **Stucture informatique.** Il serait très inefficace de parcourir l'image à nouveau à chaque fois que l'on doit simuler un point. On va donc plutôt scanner le modèle analogue 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.

- Multigrilles. Dans snesim, si l'on délaisse systématiquement des points éloignés, on risque de ne pas bien reproduire la structure de grande échelle de l'image. Pour pallier ce problème, on peut procéder par *multigrilles imbriquées*. L'idée est par exemple de ne simuler qu'un pixel à tous les « n » pixels dans un premier temps. Au départ, on ira donc chercher dans l'image de référence uniquement des états de grande échelle. Une fois la grille initiale simulée, on remplit la grille interne. On peut avoir ainsi plusieurs grilles imbriquées, habituellement selon des puissances de 2.

Ainsi si l'on veut simuler une image de 512 x 512 pixels, on peut d'abord ne simuler qu'un pixel à tous les 16, puis un à tous les 8, puis un à tous les 4, 1 à tous les 2 et finalement la grille entière. Chaque grille intermédiaire va demander son propre arbre de recherche et à chaque étape, on applique l'algorithme de base. Cette approche favorise la reproduction de la structure grande échelle au détriment de la texture petite échelle car il peut arriver que des pixels simulés tôt sur les premières grilles gênent considérablement lors des dernières étapes (correspondent à des états impossibles du modèle analogue).

- Un des grands avantages de l'approche snesim est que le conditionnement de pixels existants ne pose pas de problème particulier. Tout comme dans le SGS, ceux-ci sont simplement inclus au départ de l'algorithme.

- **Multiples faciès.** Si au lieu d'avoir 2 catégories on en a p, alors le nombre d'états possibles dans la simulation effectuée avec un voisinage de n pixels devient p^n. Ce nombre devient grand très rapidement indiquant que l'on peut difficilement appliquer l'algorithme snesim sur des catégories ou des variables discrètes lorsque p est grand. En effet dans ce cas, la probabilité de trouver au moins un état identique sur l'image de référence tend rapidement vers zéro.

Approche séquentielle par paquet (filtersim, simpat, patchwork,...)

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.

Filtersim (e.g. Wu, Boucher et Zhang, 2008[2]; Zhang, Switzer et Journel, 2006[3]):

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;

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.

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 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 ou 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 : Les points conditionnant doivent apparaître sur un point de la grille simulée. On les déplace au besoin vers le point de grille le plus proche, ce qui annule le conditionnement exact. Dans une stratégie multigrille, le déplacement peut être important. Si l'on ne fait pas le déplacement, alors on peut créer des structures à une échelle supérieure qui seront incompatibles avec les pixels connus à une échelle inférieure.

Arpat et Caers (2007)[4] 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.

---

[2] Wu, J., Boucher A., et Zhang T. 2008 A SGeMS code for pattern simulation of continuous and categorical variables: FILTERSIM . Computers & Geosciences, 34, 12, 1863-1876

[3] Zhang, Switzer, Journel. Filter-based classification of training image patterns for spatial simulation. Math. Geol. 38, 1, 63-80.

[4] Arpat, Caers, 2007. Conditional simulation with patterns. Math. Geol. 39, 2, 177-204.

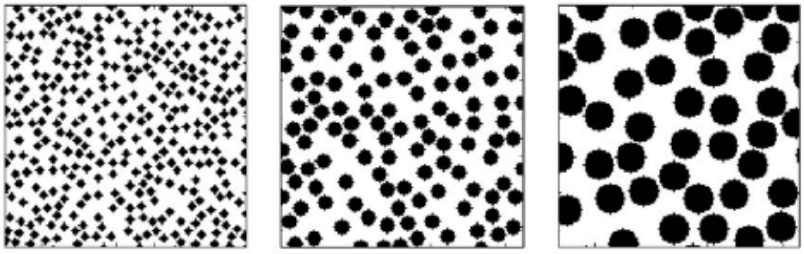
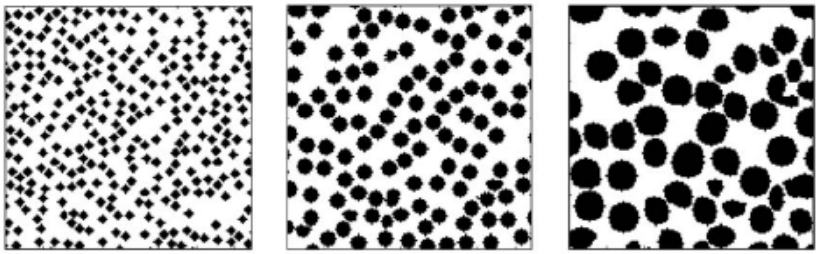
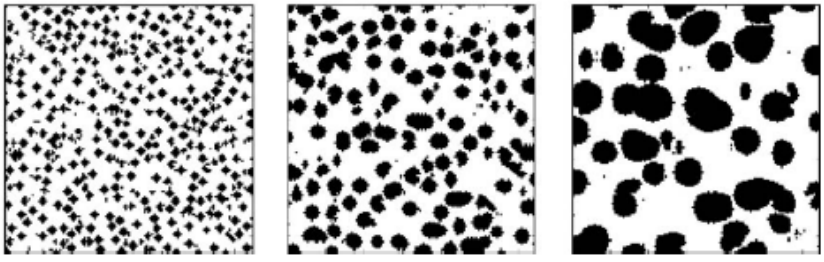
Patchwork (El Ouassani et al.[5])

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) :

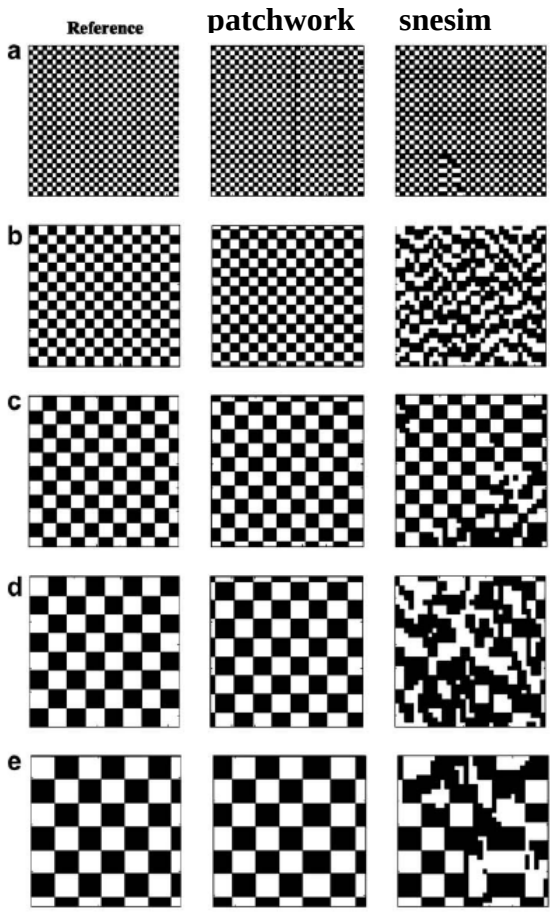
a	b
c	d

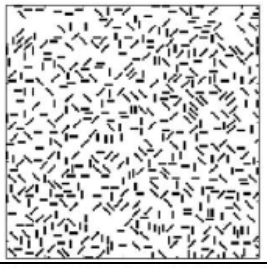
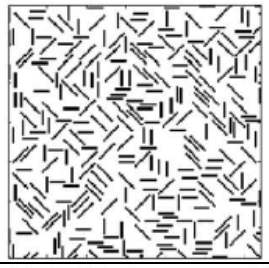
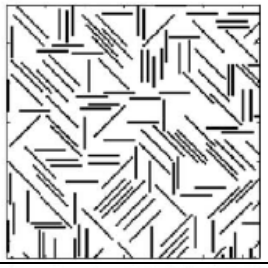
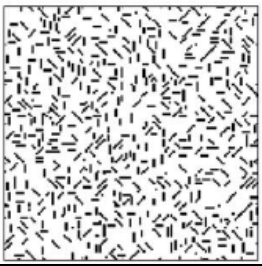
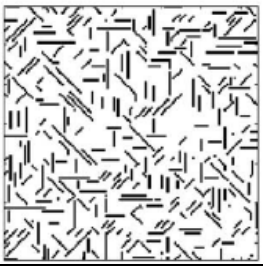
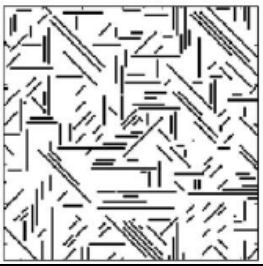
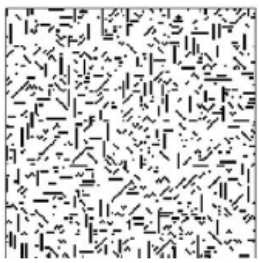
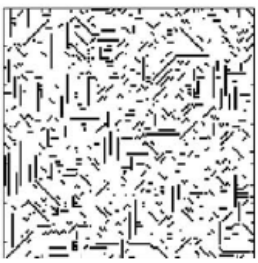
On suppose la première ligne de l'image à simuler déjà simulée ainsi que la première colonne. (pour les obtenir, on part d'une simulation de type « patchwork » avec un patron linéaire (i.e. seulement avec a, b ou seulement avec a, c) plutôt que carré). Alors, on simule d conditionnel à a, b, et c. Ensuite on fait glisser le patron d'un carré vers la droite, donc b->a, d-> c et on simule un nouveau « d » et ainsi de suite jusqu'à ce que toute l'image ait été simulée. À chaque fois, deux situations se produisent : i. soit (a,b,c) existe dans le modèle analogue et dans ce cas on choisit le « d » ou un « d » parmi les « d » associés, ii. soit (a, b, c), n'existe pas dans le modèle analogue et alors on choisit (a',b', c') le plus près de (a, b, c), en donnant dans le calcul de distance plus de poids aux pixels externes de a, b et c, de façon à maintenir le plus possible la continuité avec le reste de l'image déjà simulée. Le processus ressemble à l'assemblage d'une courtepointe, d'où le nom.

Pour inclure des données conditionnantes, on peut donner un poids très grand aux pixels correspondants dans le calcul de distance, (ceci complique toutefois l'application de l'algorithme). L'approche patchwork permet d'obtenir des textures beaucoup plus fidèles qu'avec snesim puisque l'on colle des paquets de pixels importants qui ont la bonne texture. Par contre elle ne réussit pas à bien reproduire des structures de grande échelle car elle travaille trop localement. Une amélioration possible serait d'appliquer le concept de multigrille. Cependant ceci requiert aussi un conditionnement entre échelles, ce qui vient compliquer l'application de l'algorithme et altérer la qualité de la reproduction des textures à l'échelle la plus fine, un peu comme avec snesim.

Référence	
Patchwork	
Snesim	

[5] El Ouassini A., Saucier A., Marcotte D., Favis B., 2008. A patchwork approach to stochastic simulation : a route towards the analysis of multiphase systems. Chaos, Solitons & Fractals, 36, 2, 418-436.



Référence			
Patchwork			
Snesim			

Généralisations

Simulation de variables continues

Les méthodes impliquant un calcul de distance (flitersim et patchwork) se généralisent directement au cas continu. On garde la même approche, sauf que les distances sont mesurées entre variables continues. Pour snesim la généralisation n'est pas possible.

Extension au 3D

Bien que toutes les méthodes puissent être adaptées au cas 3D, Snesim est la méthode pour laquelle cette généralisation se fait le plus facilement. Toutefois, dans tous les cas, la principale difficulté demeure la génération de modèles analogues 3D qui soient réalistes.

Données « soft »

L'intégration de données soft est la plus immédiate avec filtersim. On se sert de la variable « soft » pour informer temporairement les paquets avant de calculer les distances. La donnée « soft » vient donc favoriser la pige de certains paquets plutôt que d'autres dans le modèle analogue.

Non-stationnarité

On peut appliquer des transformations linéaires à l'image de référence (rotations, dilatations et contractions) si l'on a une information disponible à cet effet. Ceci revient à travailler localement avec des images de référence différentes. Boucher (2009)[6] propose une généralisation de cette idée avec Snesim. Il effectue une classification préliminaire du modèle analogue en patrons homogènes (basée sur des filtres appliqués localement). Chaque patron homogène est traité séparément (e.g. arbre de recherche ou liste distincts). Sur l'image à simuler, l'indicatrice de classe (par exemple avec un gaussien tronqué ou un plurigaussien) est d'abord simulée. La valeur de l'indicatrice indique quel arbre ou quelle liste utiliser pour la simulation.

Contrôle des moments d'ordres inférieurs

On observe souvent une dérive des statistiques globale d'ordre 1 ou 2 de la simulation, qui s'écartent de celles des données pour rejoindre celles de l'image de référence. Certaines tentatives ont été menées pour tenter de contrôler les statistiques d'ordre inférieur. En général, on constate qu'il y a un prix à payer. Plus on contrôle bien les statistiques d'ordre inférieur, plus on perd sur la qualité de la reproduction des textures et structures.

Problèmes rencontrés

- Le premier problème est évidemment la production de modèles analogues réalistes. En 3D particulièrement, on doit le plus souvent recourir à des simulations d'objets pour construire les modèles analogues. Il n'est pas toujours évident comment construire ces simulations.
- On observe souvent une dérive des statistiques globales d'ordre 1 ou 2 de la simulation, qui s'écartent de celles des données pour rejoindre celles de l'image de référence. La dérive de la moyenne peut être catastrophique pour des applications comme en mine par exemple. Il est un peu gênant de prétendre qu'un champ simulé représente bien la texture et la structure du phénomène représenté (l'objectif du multipoint) alors que sa moyenne s'écarte considérablement des données. S'agit-il vraiment de simulations conditionnelles? Ceci indique que toute incompatibilité entre les

[6] Boucher, A., 2009. Considering complex training images with search tree partitioning. Computers & Geosciences, 35, 1151-1158.

modèles analogues et la réalité risque de créer de réels problèmes. Faucher et al. (2012)[7] présentent une méthode pour contrôler l'histogramme de la moyenne locale.

- La reproduction des caractéristiques de l'image de référence est parfois excellente. Comme l'image de référence peut être souvent très éloignée de la réalité, doit-on s'y accrocher autant? Échantillons-nous l'ensemble des résultats possibles de façon adéquate. Devrait-on pouvoir considérer plusieurs modèles analogues? Comment les obtenir?
- Comment incorporer des informations « soft » comme l'historique de production d'un réservoir pétrolier par exemple (history matching)? La littérature est silencieuse sur ce point.

[7] Faucher C., Saucier A., Marcotte D., 2013. A new patchwork simulation method with control of the local-mean histogram. Serra, 27, 253-273.