

---

# Simulations (compléments)

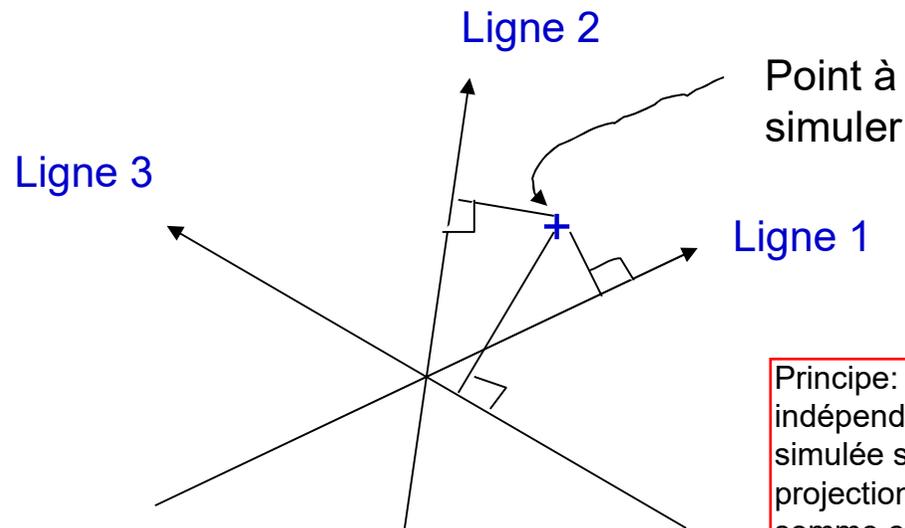
---

# Autres méthodes de simulation

## Bandes tournantes

Principe :

par projection, ramener le problème de simulation en 3D à une série de problèmes en 1D.



Principe: on simule sur chaque ligne indépendamment. On va lire la valeur simulée sur chaque ligne au point de projection du point  $x$  sur la ligne. On somme ces valeurs. Les  $Z(x)$  (en 3D) ainsi obtenus ont la covariance souhaitée si on a simulé la bonne covariance sur chaque ligne.

---

Soit :

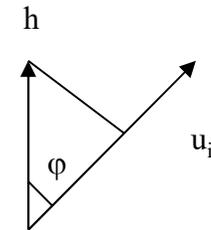
- un grand nombre ( $M > 100$ ) de lignes balayant la sphère unité et passant par le centre de la sphère;
- sur chaque ligne  $\Rightarrow$  même covariance  $C_y(h)$ ;
- sur chaque ligne  $\Rightarrow$  réalisation  $Y_i(t)$  ayant  $C_y(h)$ ;
- un point «  $x$  » à simuler définit en 3D un vecteur  $h=x$ ;
- projection de  $h$  sur  $u_i \Rightarrow Y_i(t_j)$  simulée;
- somme des valeurs trouvées sur les  $M$  lignes
- multiplication par  $(1/M)^{0.5}$  (normalisation)

en pratique une demi-sphère suffit pour couvrir toutes les directions possibles sur la sphère.

L'algorithme définit un processus  $Z(x)$  dont la covariance est :

$$C_Z(\mathbf{h}) = \frac{1}{M} \sum_{i=1}^M C_Y(\mathbf{h}'\mathbf{u}_i)$$

où  $\mathbf{u}_i$  est le vecteur unitaire définissant la  $i^{\text{ème}}$  droite;



si on laisse tendre  $M$  vers l'infini et que l'on balaye toute la demi-sphère, on aura alors (coordonnées sphériques):

$$C_Z(\mathbf{h}) = \frac{1}{2\pi} \int_0^{2\pi} \int_0^{\pi/2} C_Y(h \cos \varphi) \sin \varphi d\varphi d\theta$$

Posant  $t=h \cos(\varphi)$

$$C_Z(\mathbf{h}) = -1/h \int_h^0 C_Y(t) dt = 1/h \int_0^h C_Y(t) dt$$

changement de variable  
pour faciliter l'intégration

En inversant la relation précédente, on détermine la covariance  $C_Y(h)$  qui assure d'obtenir  $C_Z(h)$  en 3D :

$$C_Y(h) = \frac{d}{dh} [hC_Z(h)]$$

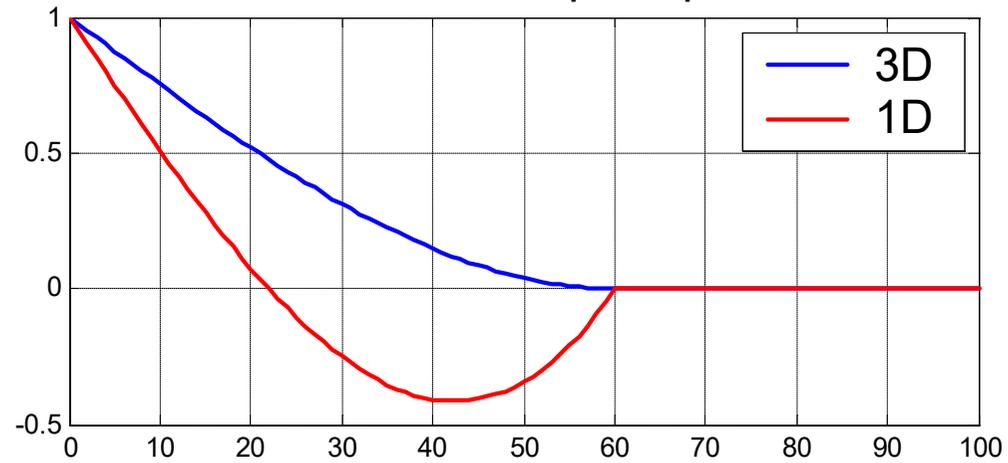
connaissant  $C_z$  que l'on veut simuler en 3D on trouve  $C_y$  que l'on doit simuler en 1D.

Exemple :

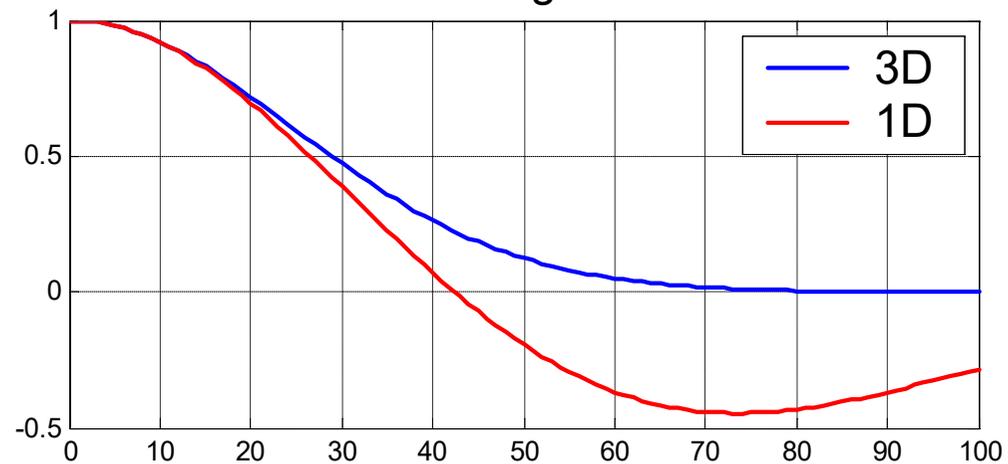
Modèle	Covariance 3D	Covariance ligne
Sphérique	$C(1-1.5h/a+0.5(h/a)^3)$ $h < a$	$C(1-3h/a+2(h/a)^3)$ $h < a$
Gaussien	$C \exp(-h^2/a^2)$	$C \exp(-h^2/a^2)(1-2h^2/a^2)$

# Exemples

## Covariance sphérique



## Covariance gaussienne



---

Simulation en 2D :

générer les lignes en 3D; placer les points à simuler en 2D

Simuler des anisotropies :

-Rotation + contraction => modèle isotrope

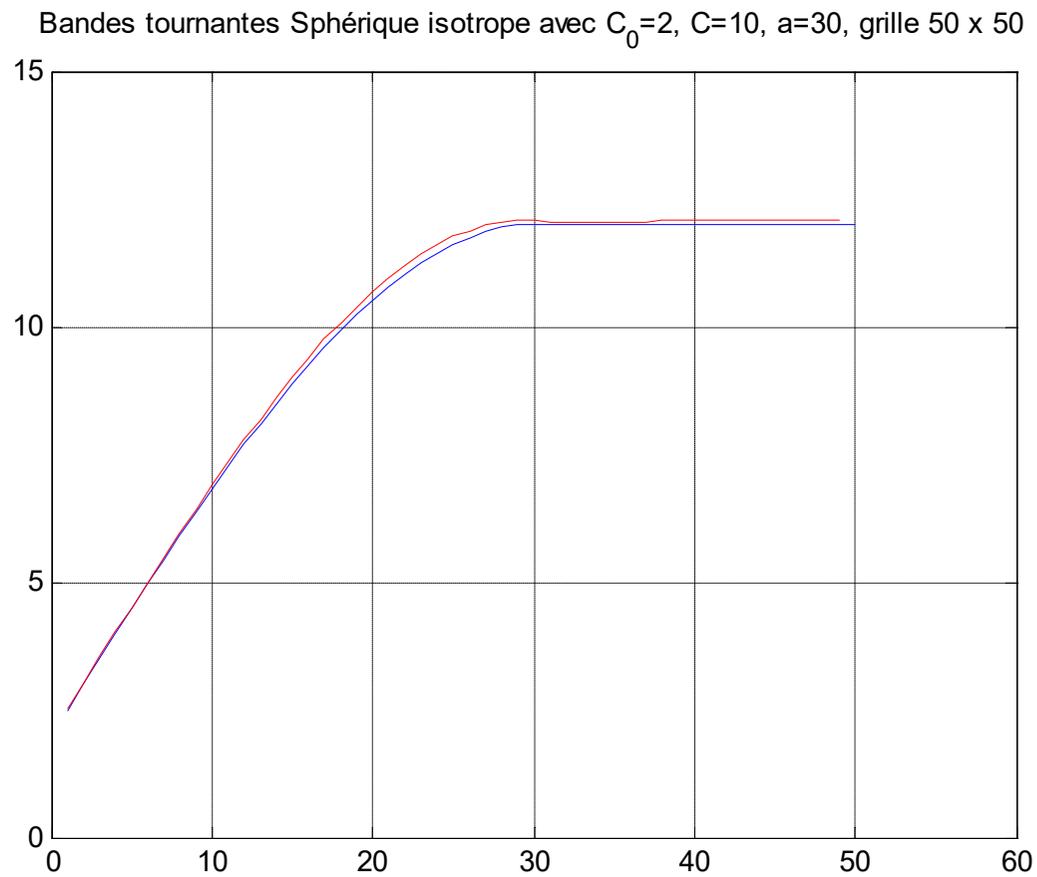
-Simuler modèle isotrope aux coordonnées transformées

-Dilatation + rotation inverse => modèle anisotrope aux coordonnées originales

Simulation conditionnelle :

Technique du post-conditionnement

Exemple (moyenne des variogrammes de 500 réalisations):



## Avantages et désavantages de la méthode :

### Avantages

-Méthode la plus efficace en espace mémoire et temps de calcul

(sans rival sérieux pour simuler des grands champs à plusieurs millions de points car complexité:  $O(n_{\text{ligne}} * n_{\text{points}}) \propto O(n_{\text{points}})$ )

### Désavantages

- Plus compliquée à programmer
- Post-conditionnement requis
- Anisotropies
- Plusieurs structures imbriquées  
=>simuler séparément chaque structure

# Méthode FFT-MA

MA est pour "moving average" donc moyenne mobile (d'un bruit blanc). La moyenne est pondérée selon la distance au point simulé

Principe :

Un champ autocorrélé =>

convolution (bruit gaussien, fonction de pondération)

$$Z = Y * g \quad \Longrightarrow \quad Z(x) = \int_{-\infty}^{\infty} Y(t)g(x - t)dt$$

Y est bruit blanc et g est une fonction de pondération symétrique par rapport à 0

Convolution =>

simple produit dans le domaine fréquentiel

$$Z = \mathfrak{F}^{-1} \mathfrak{F}(Z) = \mathfrak{F}^{-1} \mathfrak{F}(Y * g) = \mathfrak{F}^{-1} \left( \mathfrak{F}(Y) \mathfrak{F}(g) \right)$$

Fonction g ?

faut que : 
$$C(h) = g * \check{g} = \int g(t)g(t+h)dt$$

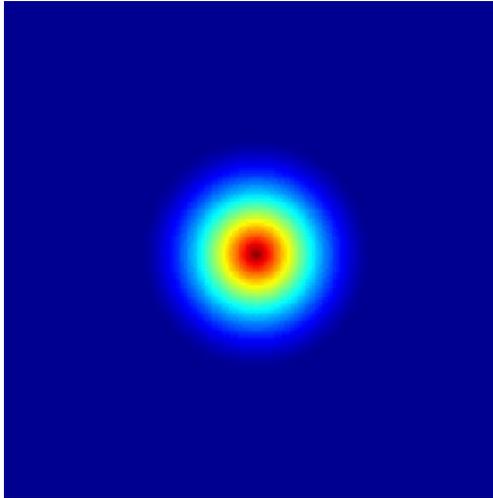
g(t) est symétrique en t,  
g=g(t) et le g marqué est g(-t)

Plusieurs choix possibles; un choix particulier :  $\mathfrak{J}(g) = \mathfrak{J}(C)^{0.5}$

g est alors symétrique

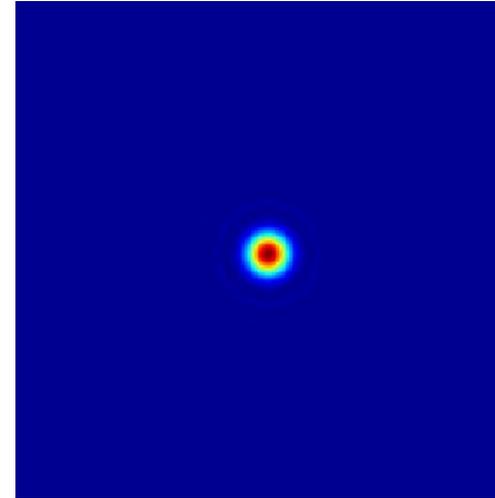
Numériquement, on prend FFT plutôt que  $\mathfrak{J}$

$C(h)$



L'origine ( $h=0$ )  
est au centre de  
l'image

$c(w)$

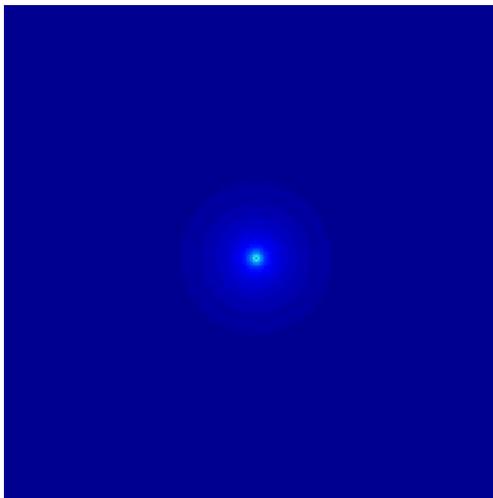


$c(w)$  est la FFT de  
 $C(h)$  (domaine des  
fréquences)

FFT

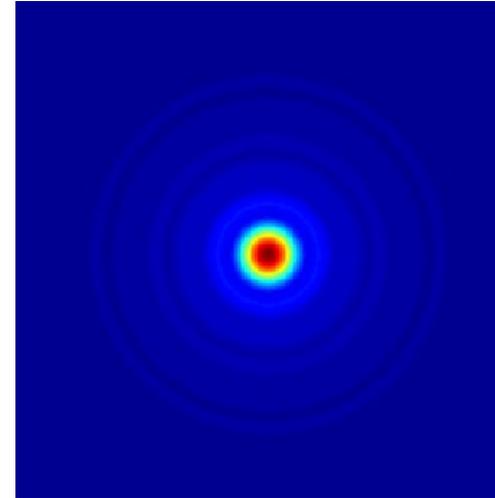


$G(h)$



fonction de  
pondération dans  
l'espace.

$g(w) = c(w)^{0.5}$

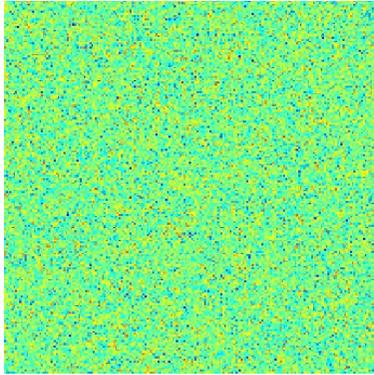


la racine carré,  
domaine des  
fréquences

FFT<sup>-1</sup>

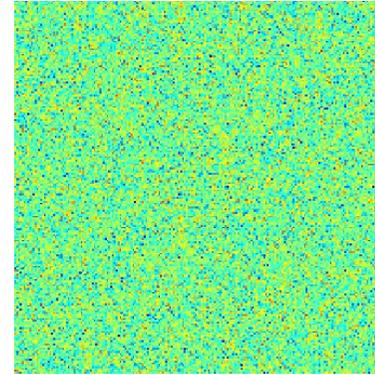


domaine à simuler  
on génère un bruit  
blanc gaussien



$Y(x)$

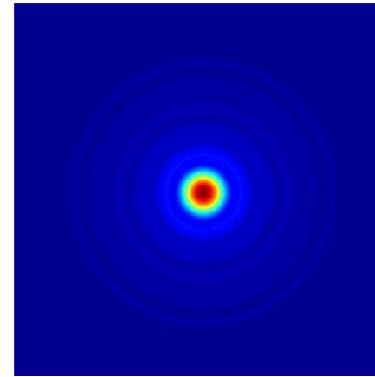
FFT



FFT du bruit blanc  
(domaine des  
fréquences)

$y(w)$

$X$

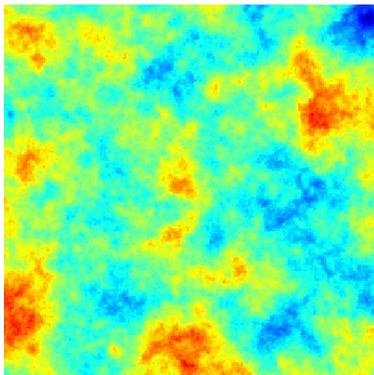


$g(w)$

fft de la fonction de  
pondération  
(domaine des  
fréquences)

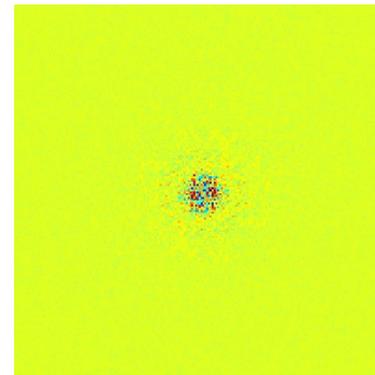
$||$

$Z(x)$



valeurs simulées  
dans le domaine  
spatial

FFT<sup>-1</sup>



$z(w)$

fft des valeurs  
simulées (domaine  
des fréquences)

Illustration en 1D : 4 points d'un modèle sphérique ayant  $C=1$  et  $a=2$ .

=> Doit simuler un champ de taille  $4+a=6$ .

C(h)	→	1	0.3125	0	0	0	0.3125
#	→	1	2	3	4	5->3	6->2

on doit symétriser la covariance

On prends la FFT de C(h)

FFT C(h)	→	1.6250	1.3125	0.6875	0.3750	0.6875	1.3125
#	→	1	2	3	4	5->3	6->2

---

On prends la racine de FFT de  $C(h) \Rightarrow$  FFT de  $g$

FFT (g)	→	1.2748	1.1456	0.8292	0.6124	0.8292	1.1456
#	→	1	2	3	4	5-→3	6-→2

On simule 6 valeurs  $N(0,1)$

Y	→	-0.4326	-1.6656	0.1253	0.2877	-1.1465	1.1909
#	→	1	2	3	4	5	6

On prend la FFT des 6 valeurs Y

FFT(Y)	-1.6407	-0.4470 +1.3724i	0.6030+ 3.5752i	-1.2667	0.6030 -3.5752i	-0.4470 -1.3724i
#	1	2	3	4	5	6

On effectue le produit FFT(g) x FFT(Y)

FFT(g) x FFT(y)	-2.0915	-0.5121 +1.5723i	0.5 +2.9644i	-0.7757	0.5 -2.9644i	-0.5121 -1.5723i
#	1	2	3	4	5	6

On prend la FFT inverse

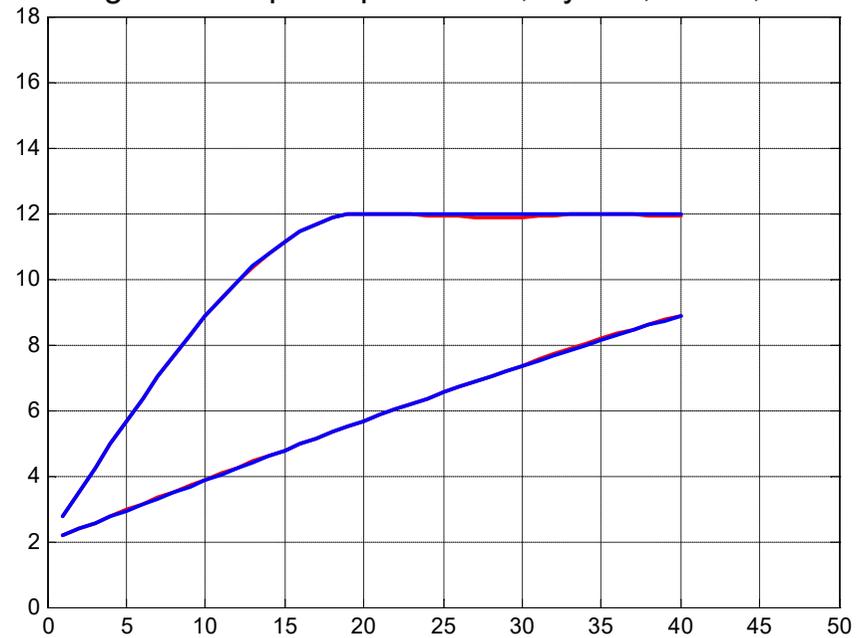
IFFT->Z	-0.4819	-1.6976	-0.0740	0.1181	-0.8777	0.9216
#	1	2	3	4	5	6

on retient quatre valeurs consécutives (au choix)

---

## Exemple: 500 réalisations champ 50x50 variogramme moyen

Variogramme sphérique  $a_x=80$ ,  $a_y=20$ ,  $C_0=2$ ,  $C=10$



## Avantages et désavantages de la méthode :

### Avantages

-Méthode très efficace en temps de calcul, moins efficace en espace mémoire

(ex.  $1 \times 10^6$  points en  $\frac{1}{4}$  sec.)

-Assez facile à programmer

-N'importe quelle covariance (anisotropies, imbriquées, etc.). Mais on ne peut simuler de très grandes portées sur une maille fine (espace mémoire).

-Peut être modifiée localement

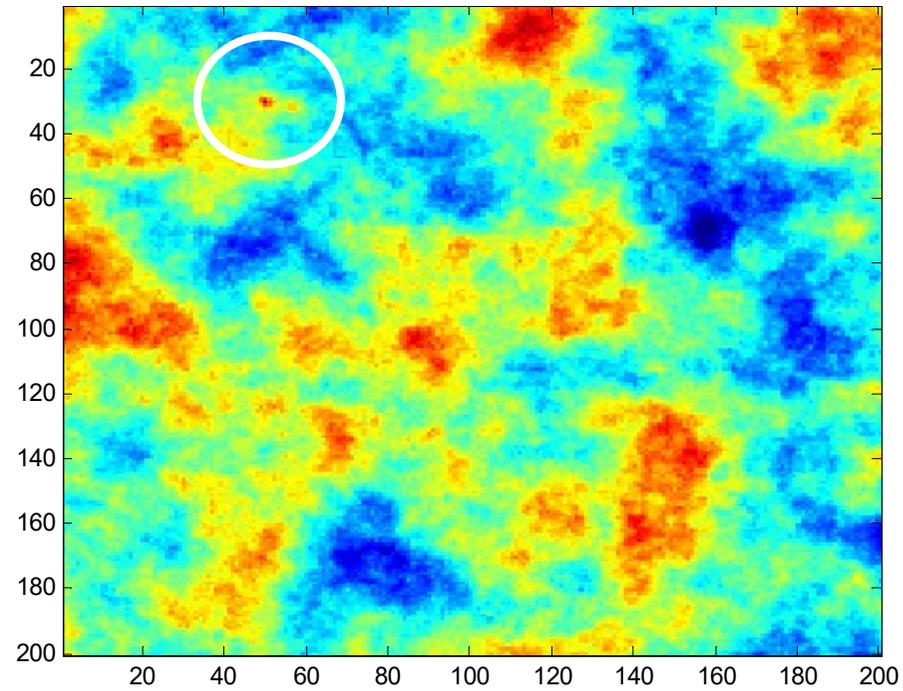
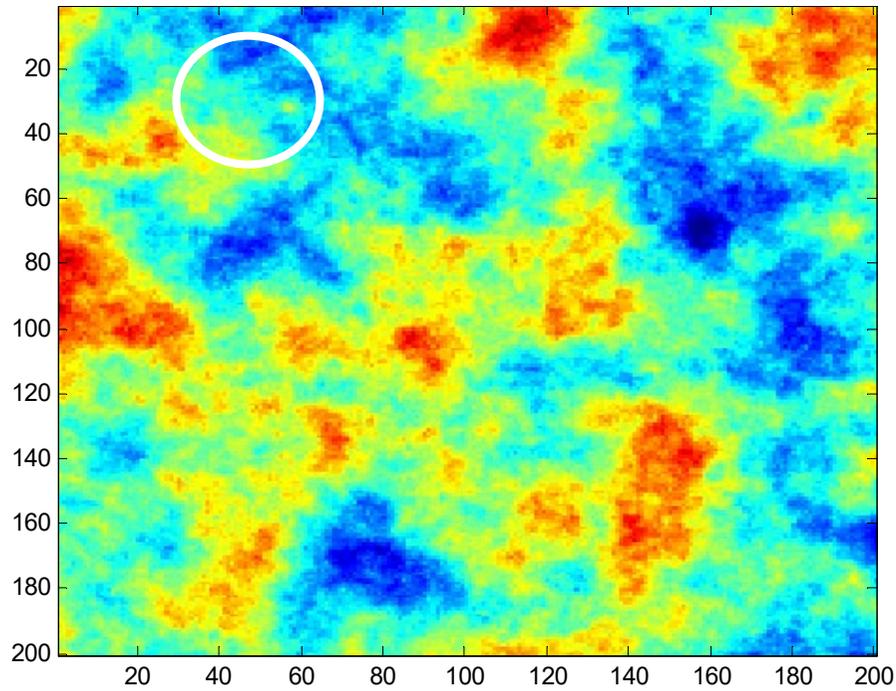
-Peut fixer l'aléa et faire varier le modèle

### Désavantages

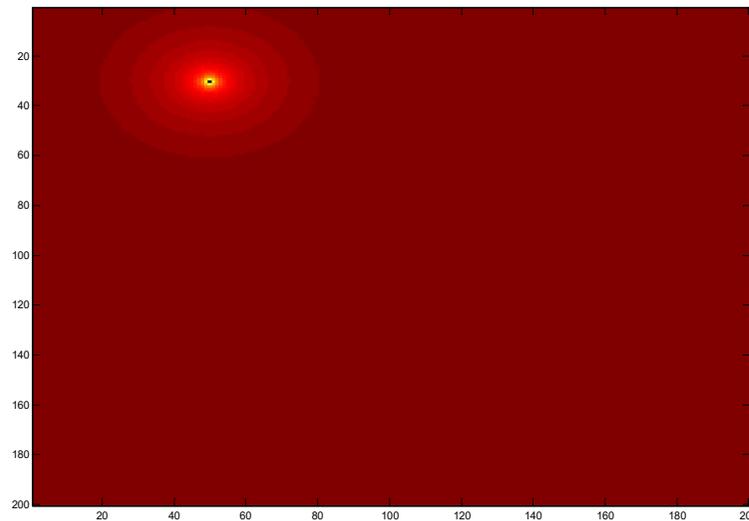
-Grille régulière seulement

-Post-conditionnement requis

On peut changer localement le terme aléatoire pour s'approcher d'un objectif (déformation graduelle)



Différence entre les 2 cartes

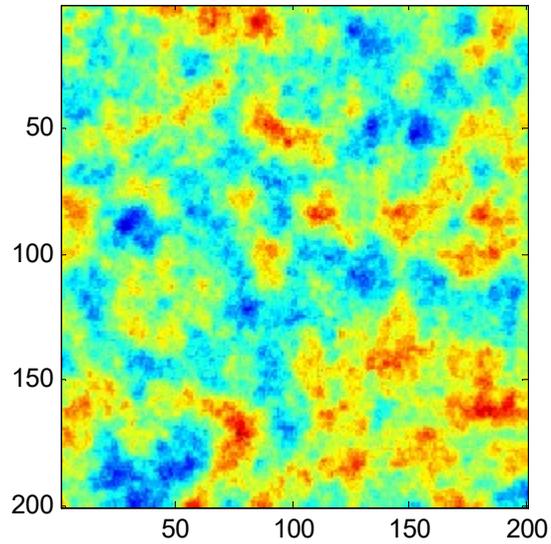


très utile pour le calage, on peut modifier localement le champ en un ou plusieurs points sans changer le reste du champ. On peut ainsi modifier pour rencontrer un objectif donné

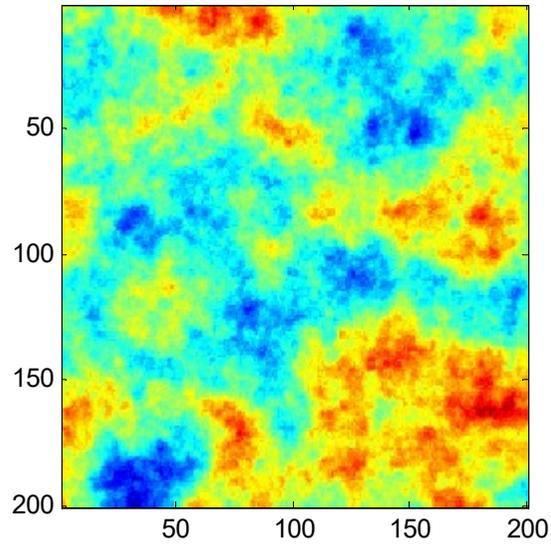
On peut fixer l'aléa et faire varier la structure seulement

ici seule la fonction de pondération change. Le bruit blanc de départ est identique.

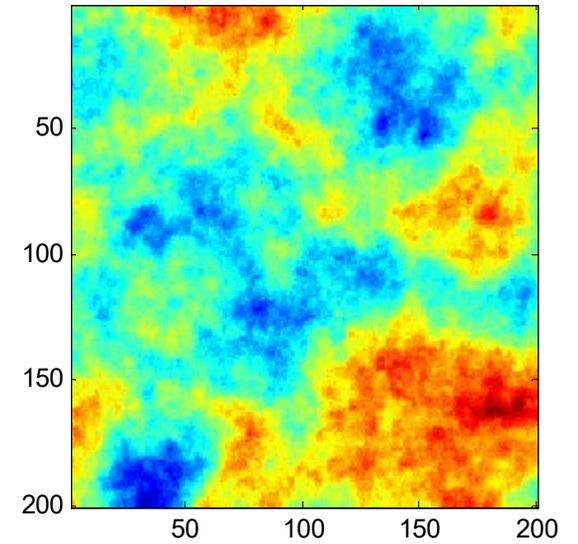
Portée a=20



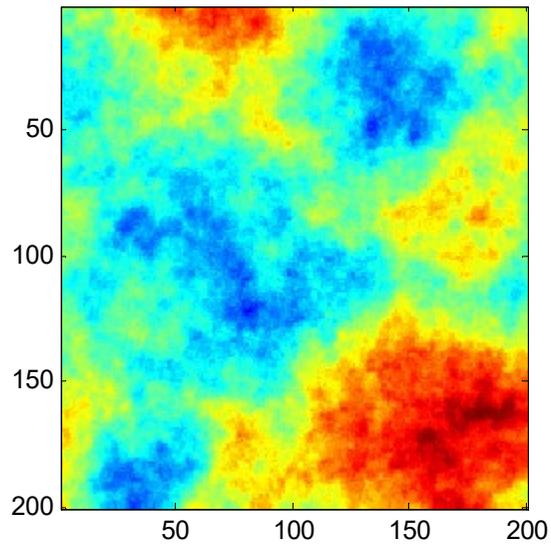
Portée a=35



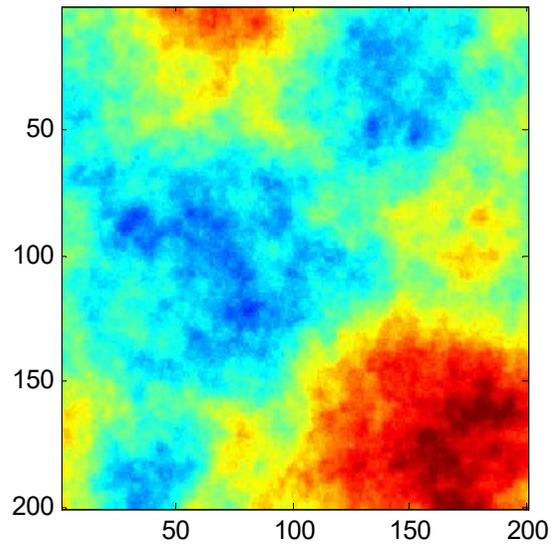
Portée a=50



Portée a=75



Portée a=100



Portée a=200

