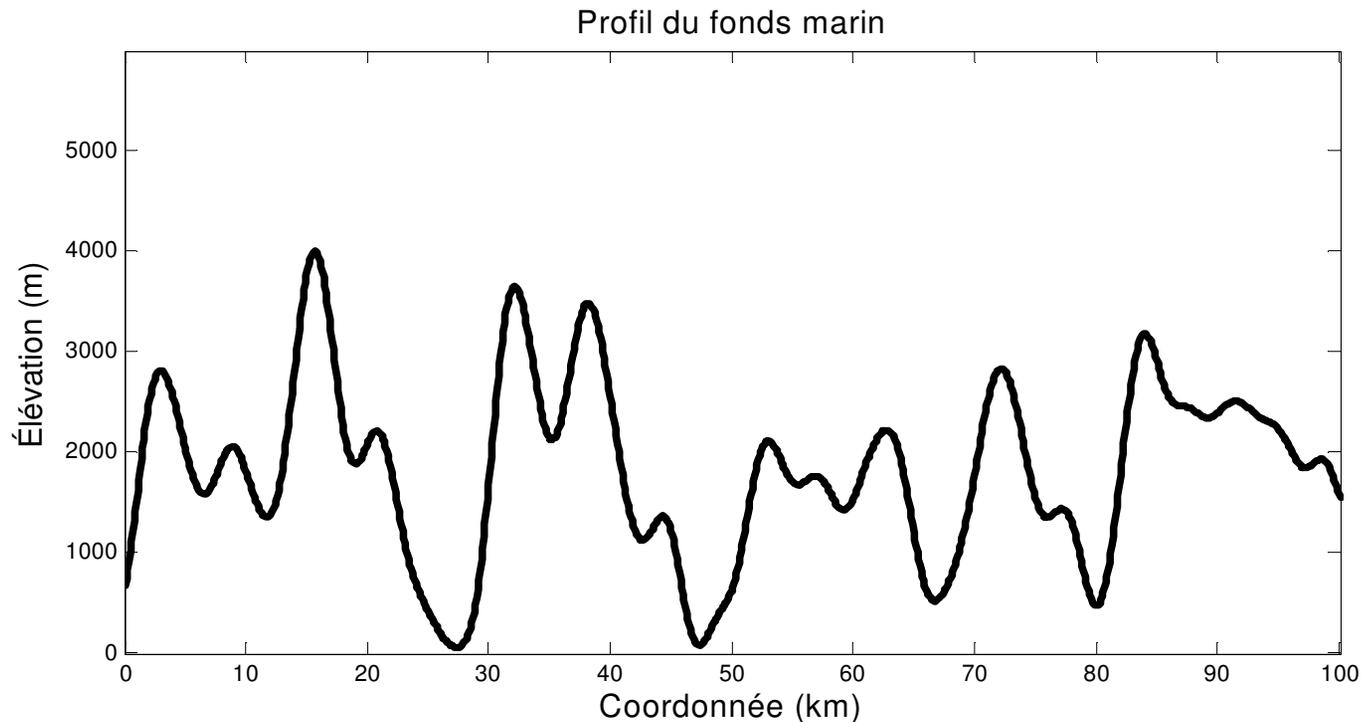

Simulations

Automne 2003

Plan

- Introduction: contexte et problématique
- Simulation vs krigeage
- Simulation non-conditionnelle vs simulation conditionnelle
- Cas gaussien
 - Méthode LU
 - Méthode SGS
- Cas non gaussien
- Simulation de teneurs de blocs
- Recuit simulé

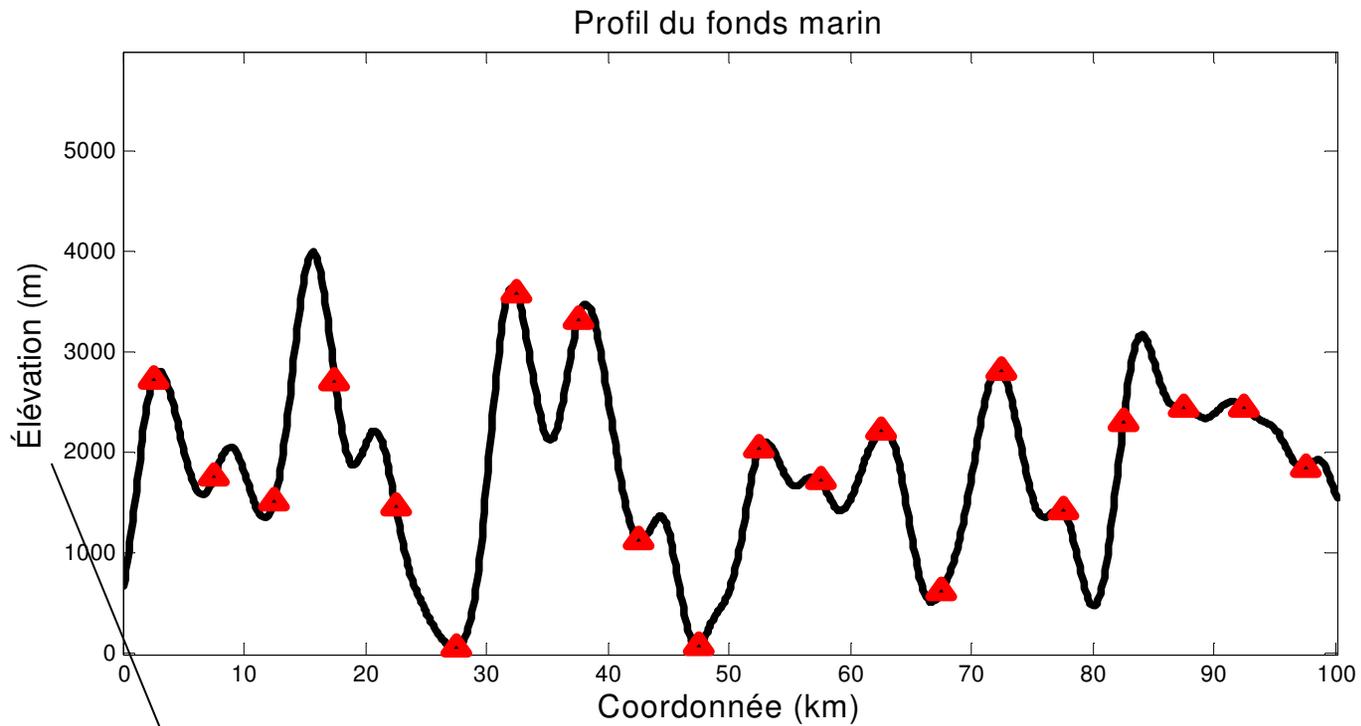
Contexte et problématique : des exemples



Vous déposez un câble de fibre optique sur le fonds marin.
Quelle longueur devrait avoir ce câble¹ ?

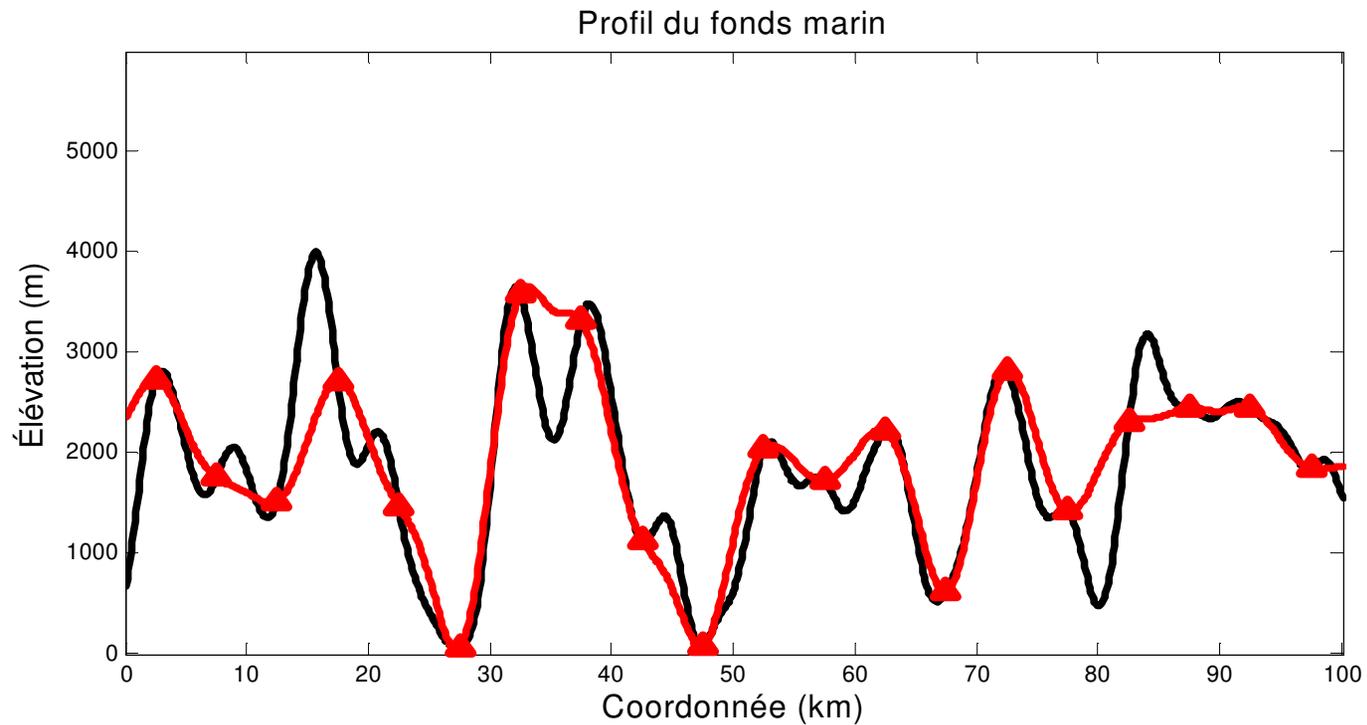
¹Exemple inspiré de Chilès et Delfiner, 1999 et Alfaro, 1979

La profondeur exacte est connue uniquement aux points observations



Exagération verticale environ 10

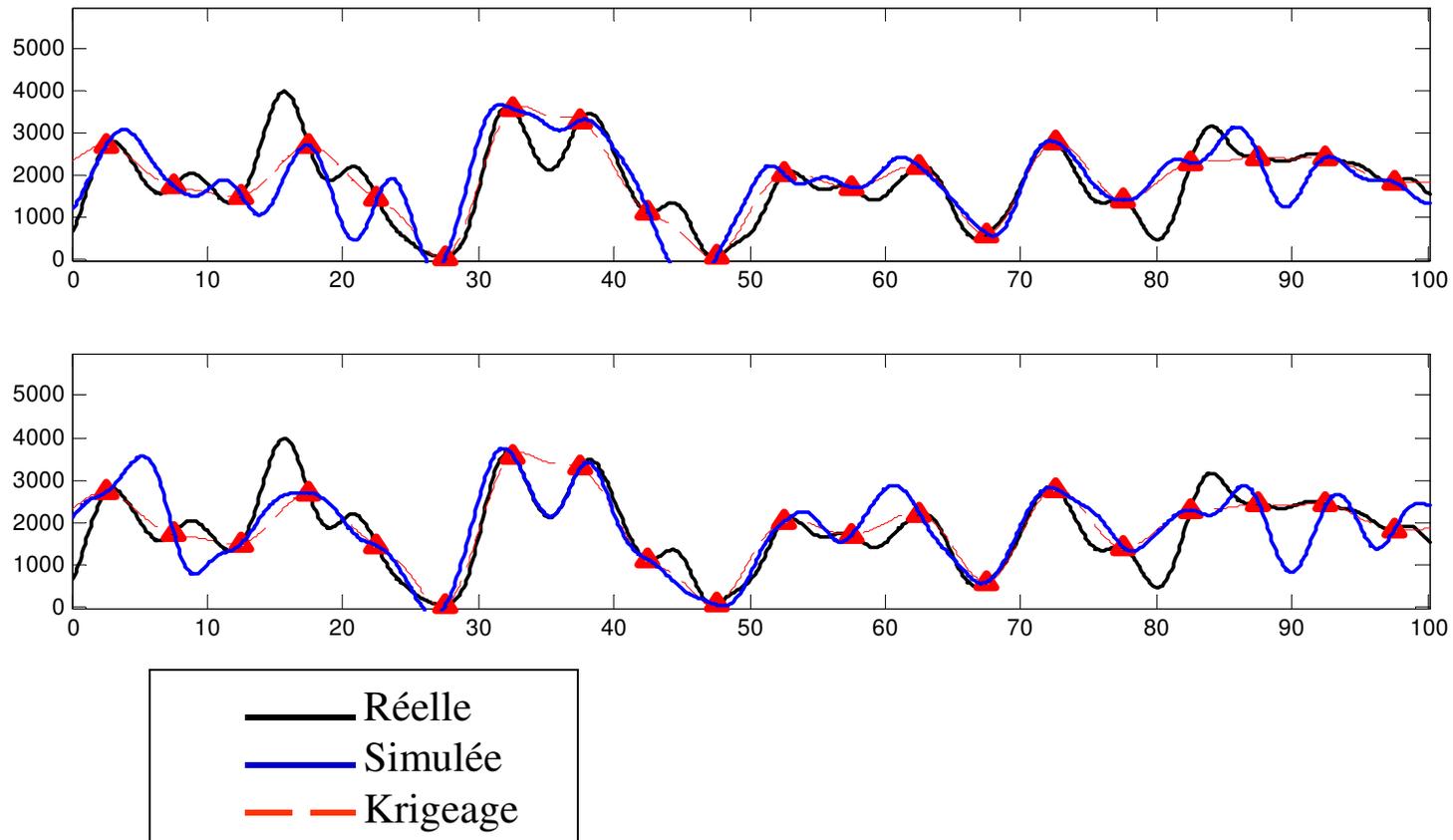
On effectue un krigeage



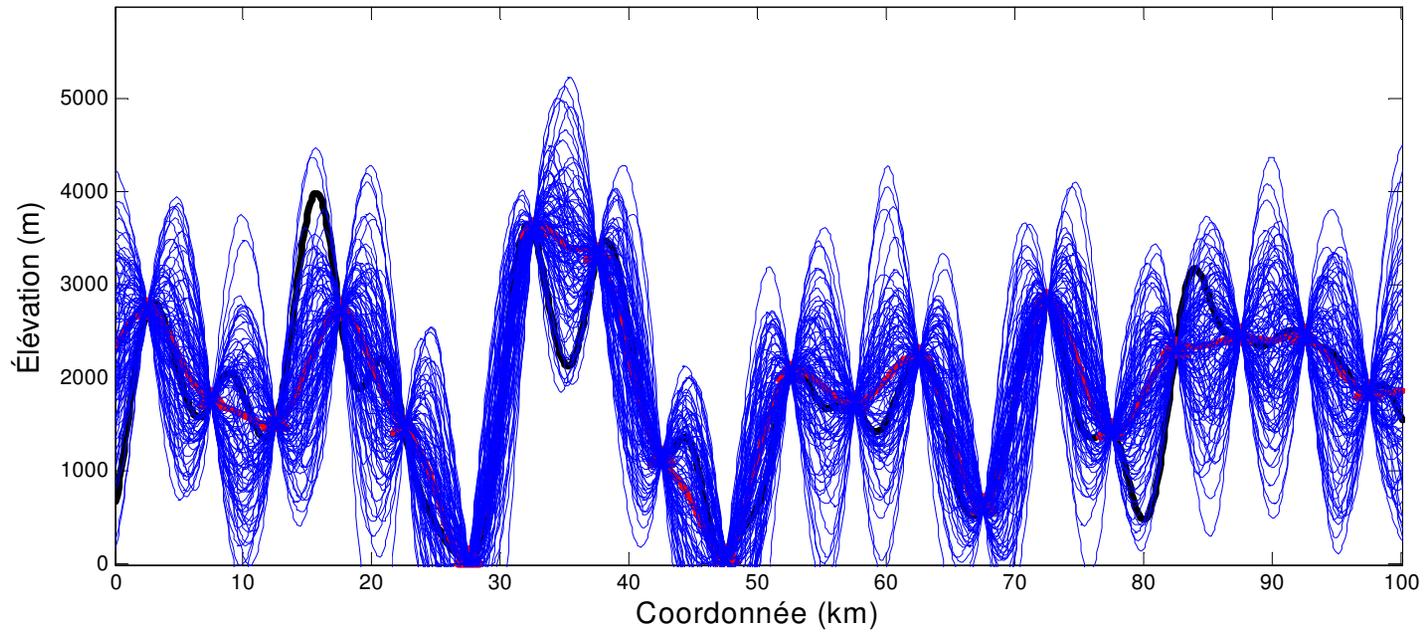
La longueur réelle est 110 km, le krigeage donne 104.6 km

On va manquer de câble !

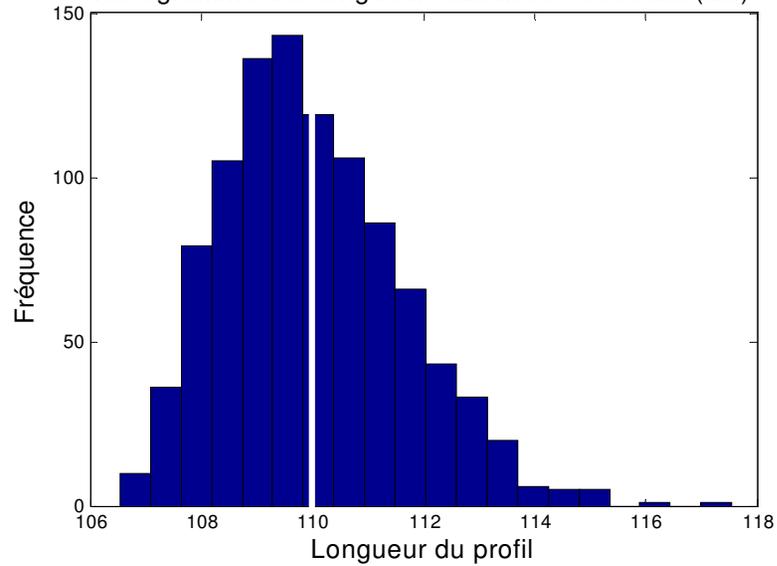
Une autre approche : des simulations conditionnelles



Profil du fonds marin



Histogramme des longueurs simulées - 1000 km (km)



Moy. 1000 réalisations: 110

Réel: 110

L'intervalle de confiance à 95% obtenu par les simulations est:

[108.8, 113.5] =>

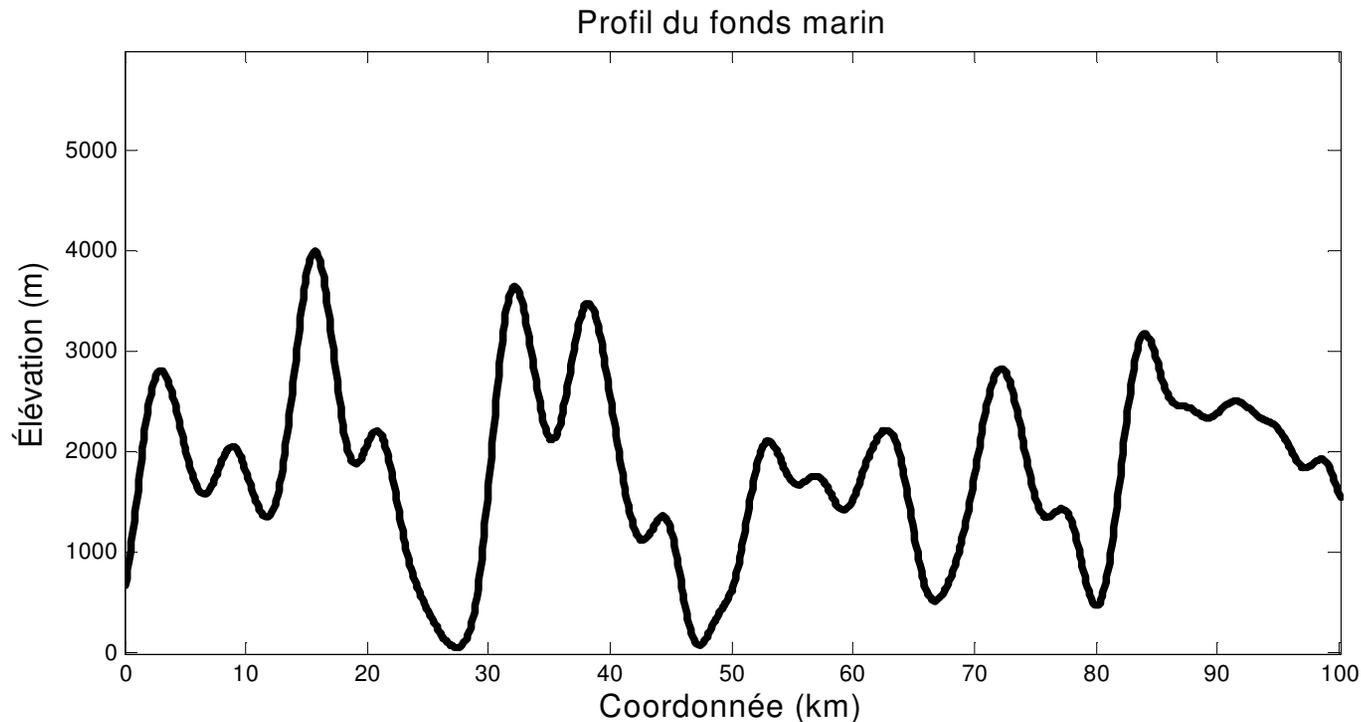
la valeur par krigeage 104.2 n'est même pas dans l'intervalle !

Longueur de câble = f (profil du fonds marin)

« f » est appelée *fonction de transfert* et

ici « f » est *non-linéaire*

Même exemple, autre fonction de transfert



Vous disposez d'un véhicule raclant le fond marin pour récolter des nodules de manganèse. Le véhicule ne peut s'accommoder de pentes supérieures à 6° . Allez-vous risquer d'utiliser le véhicule ?

Avec les mêmes données :

Pente maximale avec le krigeage: 5.8°

Pente maximale réelle : 8°

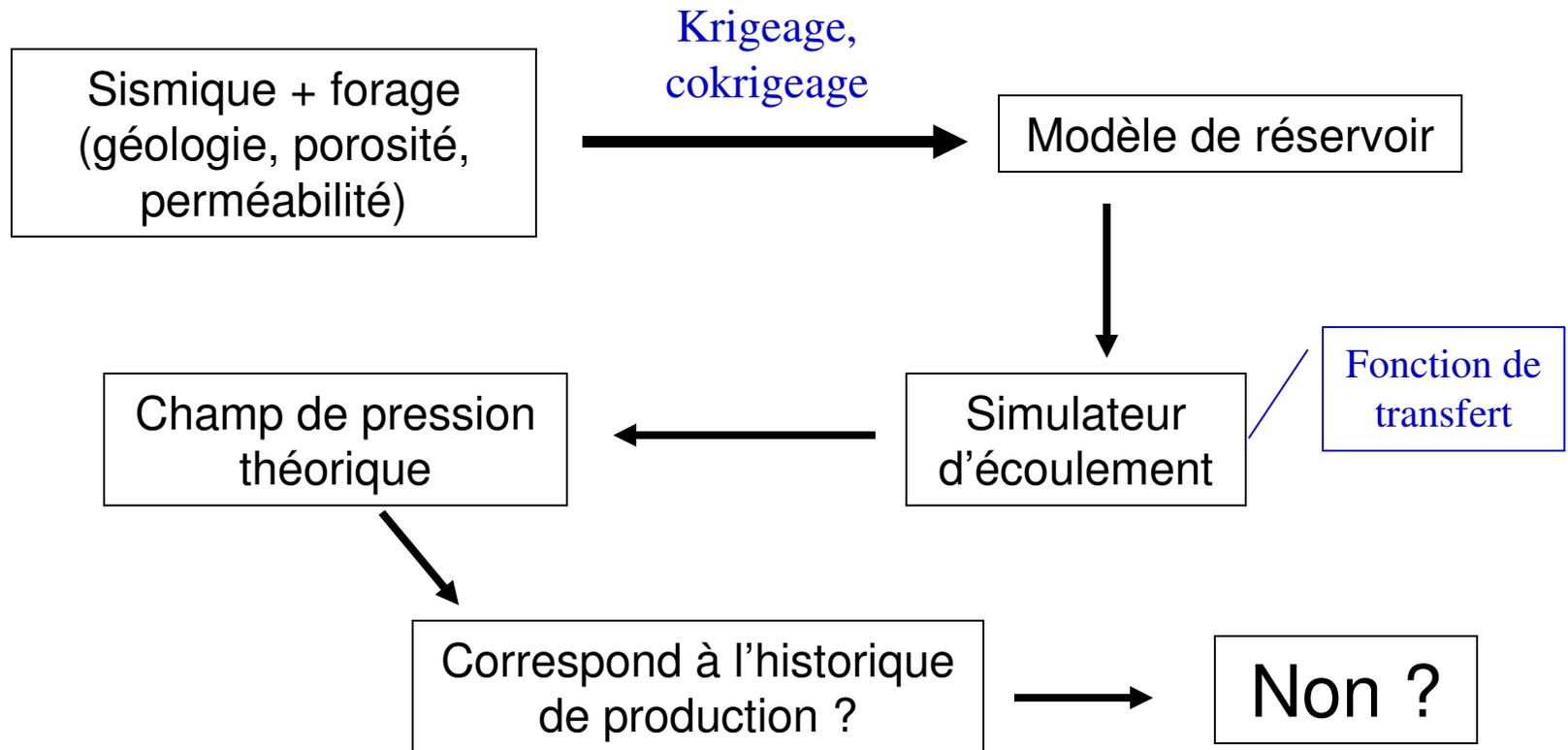
94% des 1000 réalisations montrent au moins un segment avec pente $>6^\circ$



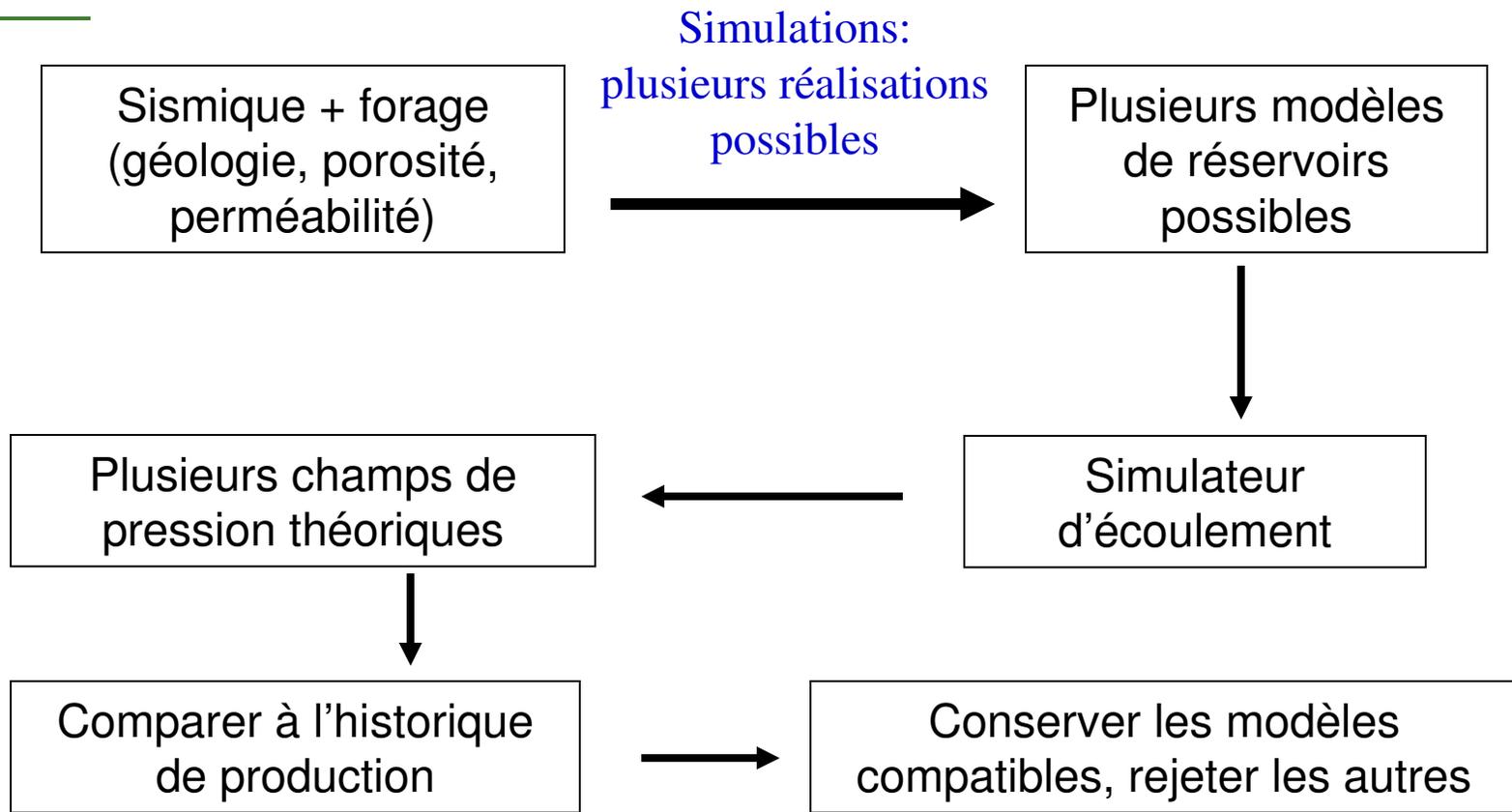
La simulation permet de reconnaître
qu'il est très risqué d'utiliser le véhicule
sur ce terrain

Quelques exemples d'application des simulations

1- Réservoir pétrolier



1- Réservoir pétrolier (suite)



Bonus : idée de l'incertitude quant au véritable modèle de réservoir

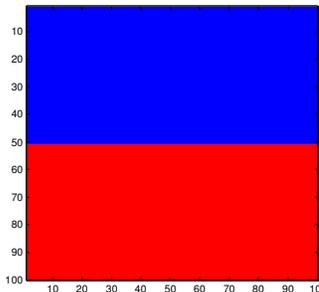
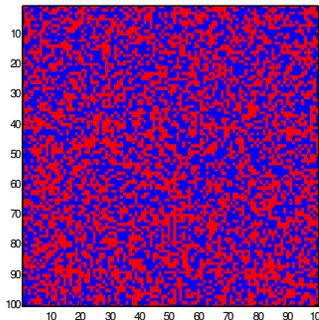
2- Hydrogéologie : changements d'échelle



Transmissivité
de bloc ?

● $T=10^{-6} \text{ cm}^2/\text{s}$

● $T=10^{-2} \text{ cm}^2/\text{s}$



$$T_{\text{bloc}} \neq T_{\text{m,point}}$$

$$T_{\text{point}} \Rightarrow \text{scalaire}$$

$$T_{\text{bloc}} \Rightarrow \text{scalaire ou tenseur}$$

$$T_{\text{m,point}} = 0.005$$

$$T_{\text{bloc}} \Rightarrow \text{moy géom} = 10^{-4}$$

$$T_{\text{m,point}} / T_{\text{bloc}} = 50$$

$$T_{\text{m,point}} = 0.005$$

$$T_{\text{bloc,horiz.}} \Rightarrow \text{moy arith} = 0.005$$

$$T_{\text{bloc,vertic}} \Rightarrow \text{moy harm} = 2 \times 10^{-6}$$

Données : tests piézométriques (quasi-ponctuel)

Simulateur d'écoulement nécessite des transmissivités de blocs (différences finies)

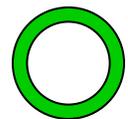
Krigeage de blocs => moyenne des transmissivités des points dans le bloc



Krigeage ponctuel + simulateur d'écoulement => les transmissivités krigées n'auront pas le bon variogramme



Simulation des transmissivités ponctuelles + simulateur d'écoulement => transmissivité de bloc



3- Hydrogéologie : aires de protection d'un puits

Données : tests piézométriques (quasi-ponctuel)

charges hydrauliques dans des puits d'observation

conditions frontières, recharge estimée, débit du (des) puits.

Simulation: plusieurs réalisations de transmissivités + simulateur d'écoulement
=> conserve réalisations qui respectent les charges observées

=> permet de dessiner plusieurs aires de captage du puits possibles

certain points du domaine seront :

- jamais captés
- toujours captés
- captés dans x% des cas



On peut aussi donner la
distribution des temps de parcours
au puits lorsque captés

La même procédure peut être suivie pour:

- déterminer des distributions de temps de transport
- déterminer les trajets possibles d'une contamination
- etc.

4- Mines : optimisation de fosses à ciel ouvert

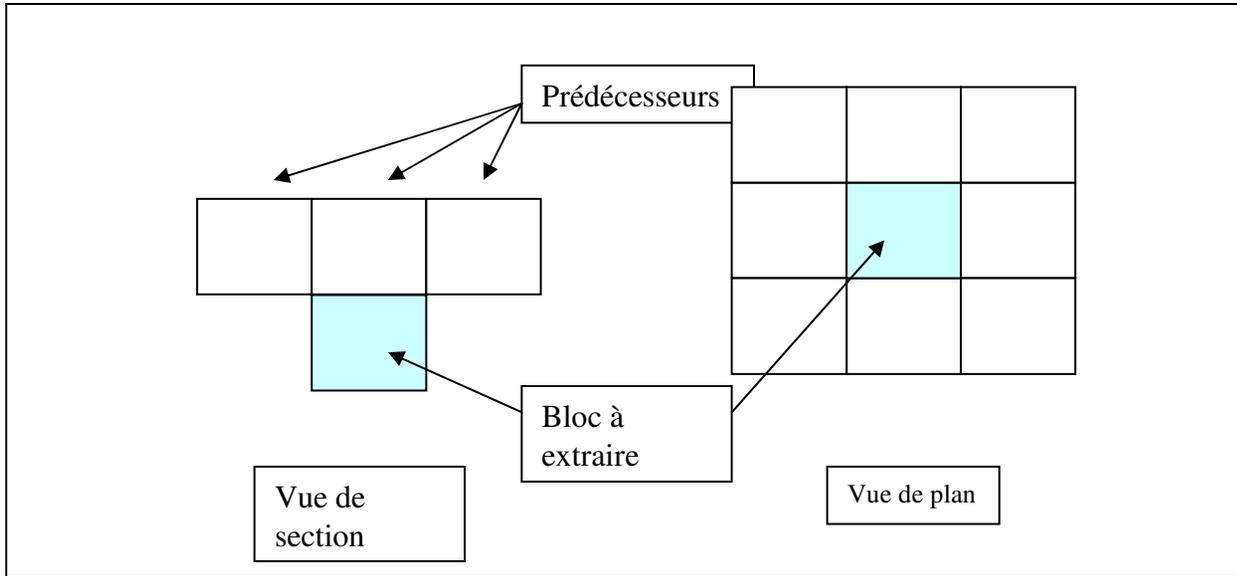
Modèle de blocs krigés => optimisation des contours de la fosse par des algorithmes d'optimisation (flow-max ou Lersch-Grossman)

Lissage du krigeage => fosse optimisée peut être très éloignée de la fosse qui sera réellement minée

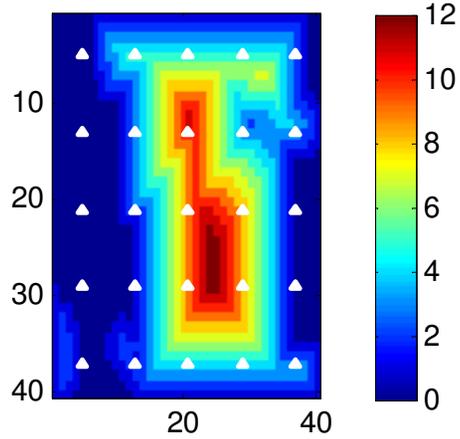
Modèle de blocs simulés => 1 fosse simulée

« n » réalisations => « n » fosses simulées

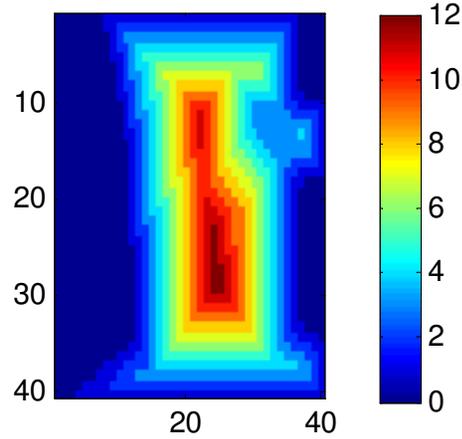
Normalement, les « n » fosses simulées vont « encadrer » la valeur de la fosse qui sera minée. Par contre la fosse « krigée » peut ne pas s'y retrouver !



Fosse optimale valeur 2224 volume 5354

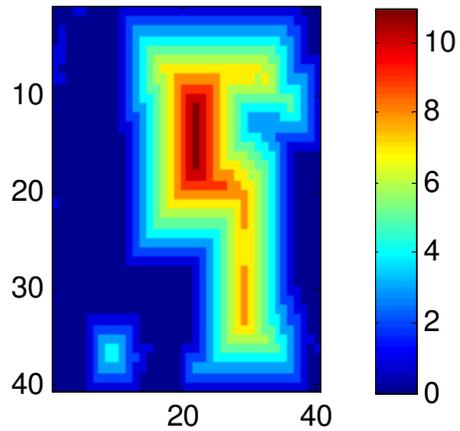


Fosse krigée valeur 2069 volume 5205

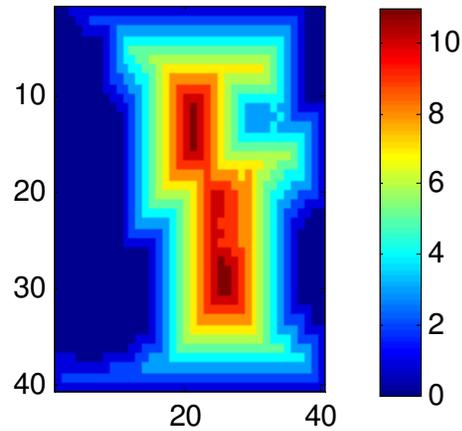


La couleur
indique le niveau
atteint dans la
fosse

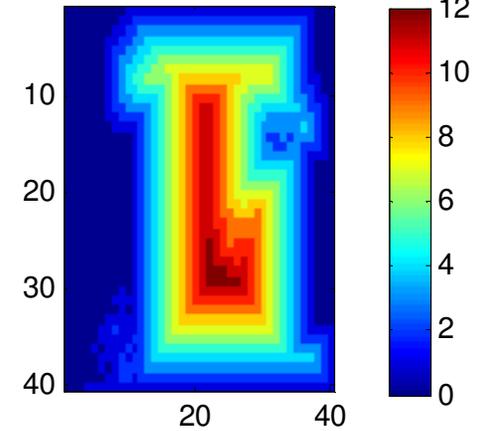
Fosse sim 72 valeur 2090 volume 4268



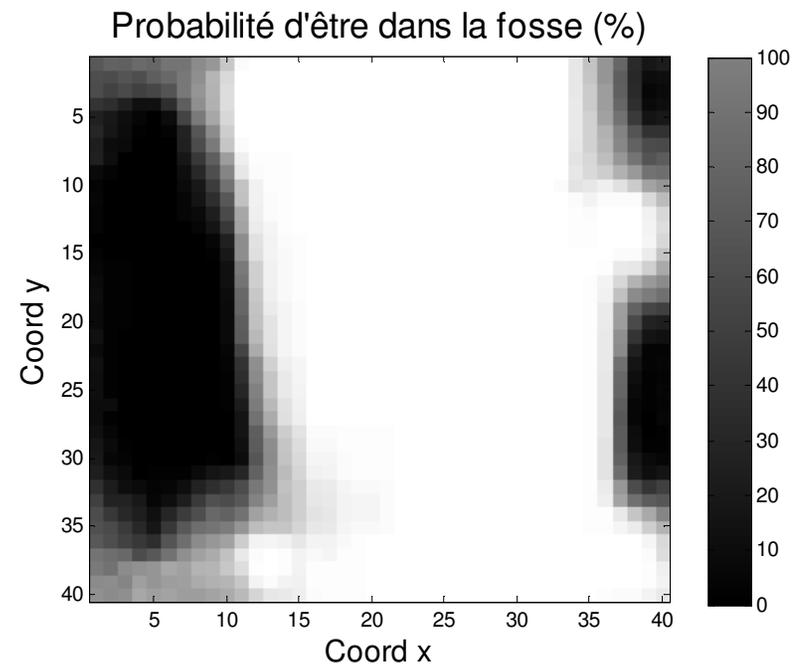
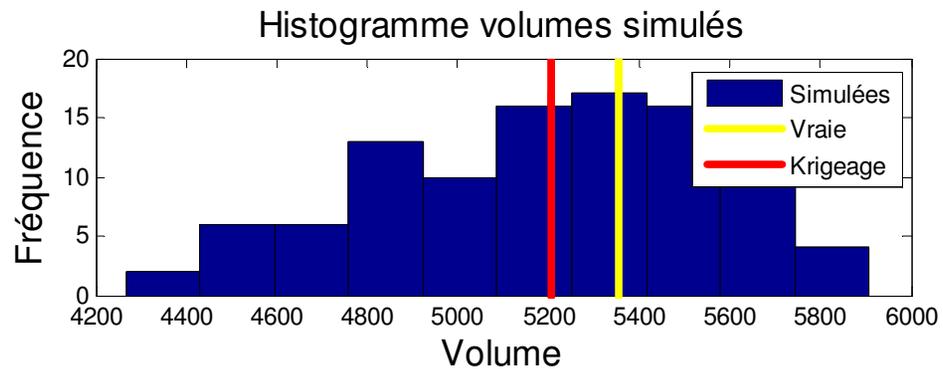
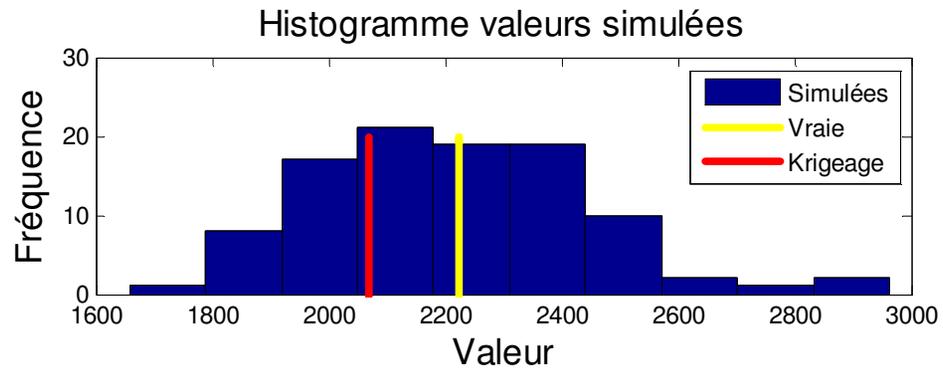
Fosse sim 34 valeur 2171 volume 5209



Fosse sim 25 valeur 1930 volume 5907



Exemple



Permet d'obtenir une mesure d'incertitude sur la valeur de la fosse, son tonnage, et sur son extension ultime possible

5- Mines : précision sur les ressources et le profit conventionnel

Chaque réalisation

=> ressources estimées différentes compatibles avec les observations actuelles

=> distribution des ressources, intervalle de confiance

Approche valide pour toute situation où l'on applique un seuil de sélection

e.g. Environnement : volume à excaver parce qu'il est contaminé

Biologie : aires avec une biomasse suffisante pour nourrir un prédateur

6- Ciment : calibration d'un logiciel de contrôle de procédé

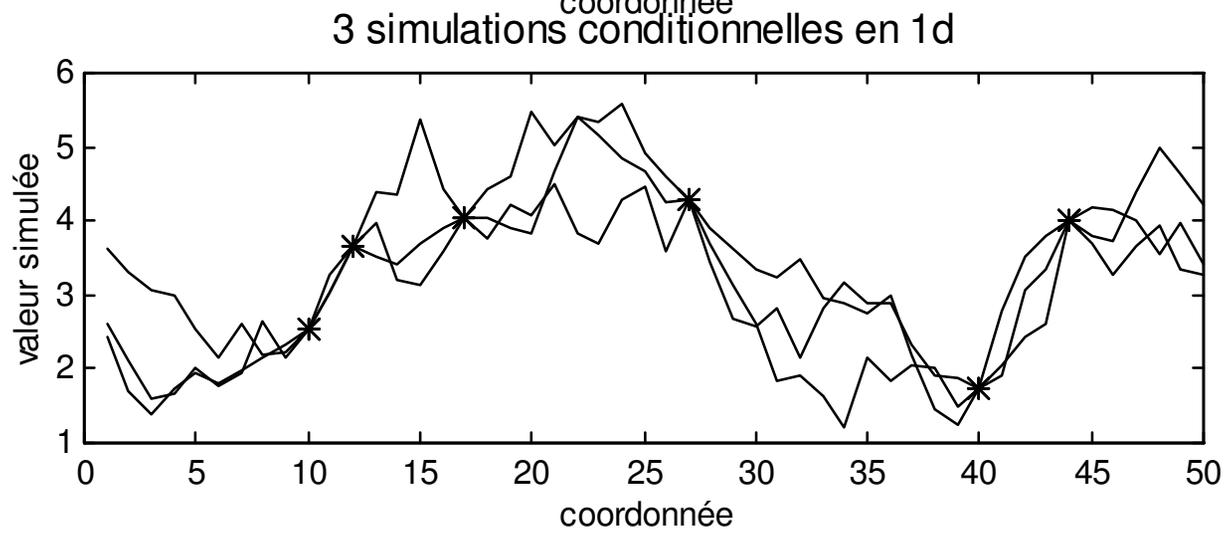
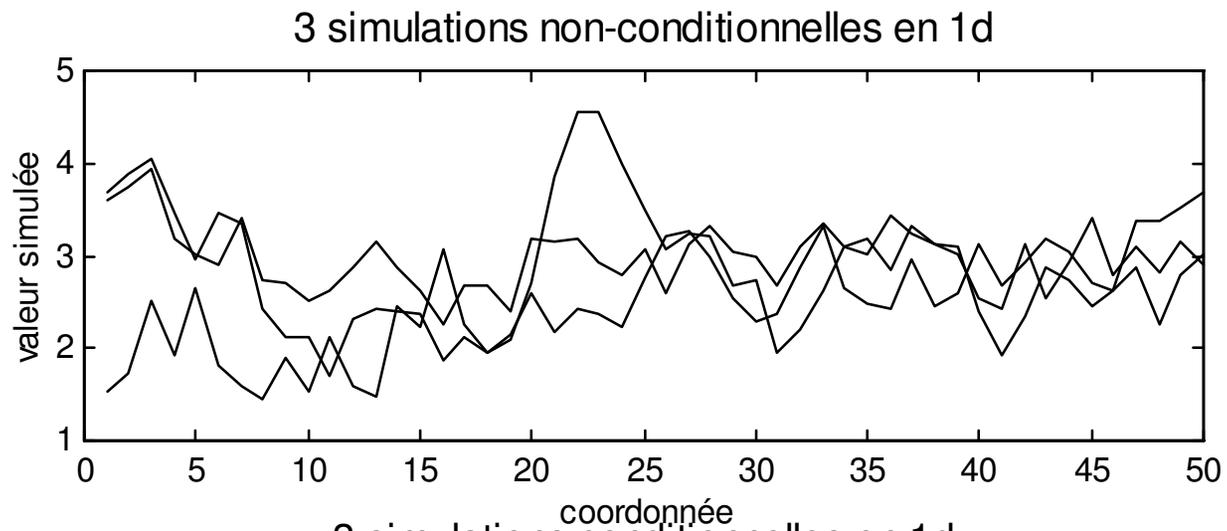
Les cimenteries utilisent des logiciels de contrôle sophistiqués pour atteindre des cibles fixes de composition du ciment.

- Plusieurs sources d'alimentation (carrière, bauxite, boues industrielles ou municipales, sable...)
- Le logiciel contrôle la vitesse d'alimentation de chaque source
- Chaque source est multivariable (8 éléments majeurs + CO₂)
- La cible est multicritère (C3S, C2S, Al₂O₃...)

- Comment régler les paramètres du logiciel de contrôle (tolérance en écart (pour quelle durée) vis à vis le critère; vitesse de correction à adopter; ...) de façon optimale ?
 - => Générer par simulation multivariable un signal d'entrée pour chaque source et étudier la performance du logiciel de contrôle en fonction des paramètres du logiciel.

Simulation non-conditionnelle et conditionnelle

Reproduit ?	Non-conditionnelle	Conditionnelle
Histogramme	oui	oui
Variogramme	oui (théorique)	oui (mélange théorique- expérimental)
Données	non	oui
Fonctions de transfert non-linéaires	Oui, mais pas directement	Oui, mais pas directement



Méthodes de simulation

Très grand nombre; ce qui les distingue :

- objets vs pixels
- gaussien vs distribution quelconque
- simulation conditionnelle possible ou non (directement)
- limité à 1D ou non
- grille régulière ou quelconque

Méthode LU (Cholesky)

1- Simulation non-conditionnelle :

Soit $Z(x)$ gaussien de moyenne 0 et de covariance $C(h)$

n points à simuler à des emplacements $x_i, i=1 \dots n$;

- i. Construire K (matrice $n \times n$ des covariances entre les n points)
 K est positive définie
- ii. Effectuer la décomposition $K=LL'$ L est triangulaire inférieure ($n \times n$)
- iii. Générer $Y_{n \times 1}$ tiré d'une $N(0,1)$ (les Y_i sont i.i.d.)
- iv. Calculer $Z=LY$

On vérifie que $E[ZZ'] = E[LYY'L'] = LE[YY']L' = LIL' = LL' = K$

2- Simulation conditionnelle :

Soit $Z^1(x)$ gaussien de moyenne 0 et de covariance $C(h)$

$Z^1(x)$ a été observé et vaut $z_i^1, i=1 \dots N$

n points à simuler en $x_j, j=1 \dots n$; conditionnellement à $z_i^1, i=1 \dots N$

- i. Construire K (matrice $(N+n) \times (N+n)$ des covariances entre les $(N+n)$ points)
 K est positive définie
- ii. Effectuer la décomposition $K=LL'$ L est triangulaire inférieure
- iii. Partager K et L en 4 blocs :

$$\left. \begin{aligned} \mathbf{K} &= \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} & \mathbf{L} &= \begin{bmatrix} \mathbf{L}_{11} & \mathbf{0} \\ \mathbf{L}_{21} & \mathbf{L}_{22} \end{bmatrix} \end{aligned} \right\} \begin{aligned} &\mathbf{K}_{11} \text{ et } \mathbf{L}_{11} \text{ sont } N \times N \\ &\mathbf{K}_{22} \text{ et } \mathbf{L}_{22} \text{ sont } n \times n \end{aligned}$$

On a toujours $Z=LY$ mais ici z^1 est observé \Rightarrow

$$z^1=L_{11}y^1 \Rightarrow y^1=L_{11}^{-1}z^1$$

Les N premières valeurs du vecteur Y sont y^1 , les « n » valeurs suivantes sont tirées indépendamment d'une $N(0,1)$.

On peut écrire :

Cette matrice n'a à être calculée
qu'une fois peu importe le
nombre de réalisations

$$z^2=L_{21}y^1 + L_{22}y^2 = (L_{21} L_{11}^{-1}) z^1 + L_{22}y^2$$

Qui fait apparaître la dépendance de z^2 sur z^1 . z^2 est le vecteur des valeurs simulées conditionnellement à z^1

Exemple

Simuler 2 variables Z_1 et Z_2 de variances 9 et 4 et de corrélation 0.8

$$\text{Cov}(Z_1, Z_2) = 0.8 (9 \cdot 4)^{0.5} = 4.8$$

$$K = \begin{bmatrix} 9 & 4.8 \\ 4.8 & 4 \end{bmatrix}$$

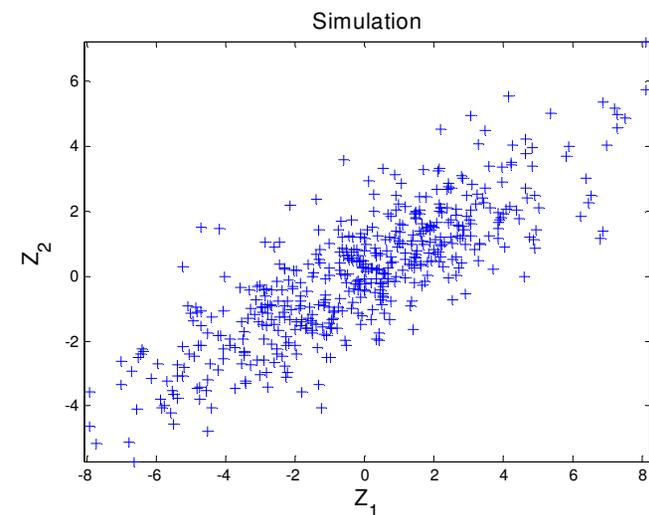
$$L = \begin{bmatrix} 3 & 0 \\ 1.6 & 1.2 \end{bmatrix}$$

Si l'on tire les valeurs :

$$Y = [-0.12 \ -0.5]' \Rightarrow Z = [-0.36 \ -0.79]'$$

$$Y = [1.47 \ 1]' \Rightarrow Z = [4.41 \ 3.55]'$$

....



Avantages de la méthode LU

- Facile à comprendre et à programmer
- Très rapide pour de petits champs
- Se généralise immédiatement au cas multivariable
- Cas conditionnel pose aucun problème.

Désavantages de la méthode LU

- Limitée à la simulation de petits champs car K doit être stockée en mémoire ($n+N$) < 10000
- K grand : K peut n'être pas positive définie numériquement (K l'est théoriquement.)
=> empêche d'effectuer la décomposition de Cholesky.

Digression : comment tirer d'une $N(0,1)$?

Plusieurs méthodes; la + courante

Tirer « p » d'une Uniforme(0,1); « p » est une probabilité.

$Z=F^{-1}(p)$ où F est la fonction de répartition de la $N(0,1)$.

Comment tirer d'une $U(0,1)$?

Plusieurs méthodes; une très courante

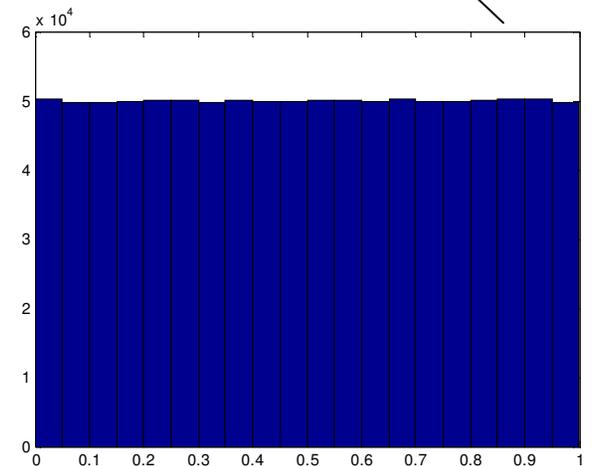
Méthode congruentielle : « a » et « m » deux entiers

$$x_{n+1} = \{a x_n \text{ mod}(m)\}$$

$$u_{n+1} = x_{n+1}/m$$

x_0 = germe entier compris entre 1 et $m-1$

Histogramme,
 $n=100,000$



Exemple : $m=2001179$, $a=1000$, $x_1=915$

n	x_n	$a x_n$	$x_{n+1} = \text{mod}(a x_n, m)$	$u_{n+1} = X_{n+1}/m$
1	915	915e03	915000	0.45723
2	915e03	915e06	461197	0.23046
3	461197	461197e03	925830	0.46264
4	925830	925830e03	1285302	0.64227
5	1285302	1285302e03	545082	0.27238
6	545802

Méthode séquentielle gaussienne (SGS)

Soit Z_i , $i=1 \dots n$, des observations d'une loi gaussienne de moyenne 0 et de covariance $C(h)$

Algorithme:

- i. Choisir un point à simuler au hasard
- ii. Effectuer le krigeage simple à ce point en utilisant les observations Z_i , $i=1 \dots n$
- iii. On tire une valeur d'une $N(Z_{KS}^*, \sigma_{KS}^2)$ (le KS, dans le cas gaussien, correspond aux paramètres de la loi de distribution conditionnelle)
- iv. Ajouter cette valeur aux valeurs observées et retourner à i.

Note: quand $n=0$, on amorce l'algorithme en tirant d'une $N(0, \sigma^2)$

SGS (démonstration par induction)

Supposons que l'algorithme permet de simuler « n » valeurs normales de moyenne 0 et de covariance $C(h)$.

On a :
$$\text{Cov}(Z_n^s, Z_n^{s'}) = \text{Cov}(Z_n, Z_n') = K_{nn}$$

$$\lambda = K_{nn}^{-1}k$$

$$k = \text{Cov}(Z_n, Z_{n+1})$$

$$Z_{n+1}^{s*} = Z_n^{s'} \lambda$$

$$Z_{n+1}^s = Z_{n+1}^{s*} + e$$

Équations du KS

e tiré d'une
 $N(0, \sigma_{KS}^2)$

$$\text{Cov}(Z_n^s, Z_{n+1}^s) = \text{Cov}(Z_n^s, Z_{n+1}^{s*}) = \text{Cov}(Z_n^s, Z_n^{s'}) K_{nn}^{-1} k = k$$



e n'est pas corrélé à Z_n^s



K_{nn}



λ

Bonne
covariance

$$\text{Var}(Z_{n+1}^s) = \text{Var}(Z_{n+1}^{s*}) + \sigma_{ks}^2 = \sigma^2$$



Relation de lissage du
KS (voir chap. 5)

Bonne variance

Note:

Au fur et à mesure que l'algorithme progresse, le nombre de points disponible pour le krigeage augmente; pour N grand, devient prohibitif.

⇒ Effectuer les krigeages en voisinages glissants (effet d'écran)

- assure approximativement la reproduction de K , d'autant mieux que l'effet d'écran est important

- certaines covariances sont difficiles à reproduire par cet algorithme (ex. modèle gaussien) car dans ce cas l'effet d'écran est faible.

Avantages de la méthode SGS

- Facile à comprendre et à programmer
- Se généralise au cas multivariable (CoKS au lieu de KS)
- Cas conditionnel immédiat

Désavantages de la méthode SGS

- certaines covariances difficiles à bien simuler
- assez lent pour de grands champs

Question

Le SGS consiste à tirer une valeur de la distribution conditionnelle obtenue par krigeage simple dans le cas gaussien

Le krigeage d'indicateur permet d'estimer une distribution conditionnelle dans le cas où la variable n'est pas gaussienne

Comment pourrait-on combiner ces 2 algorithmes pour développer une méthode séquentielle pour le cas non-gaussien ?

Cas non-gaussien

-LU et SGS nécessitent un processus $Z(x)$ gaussien

Que faire si ce n'est pas gaussien ?

i. Transformer $Y(x)=f^{-1}(Z(x))$

ii. Calculer et modéliser le variogramme de $Y(x)$

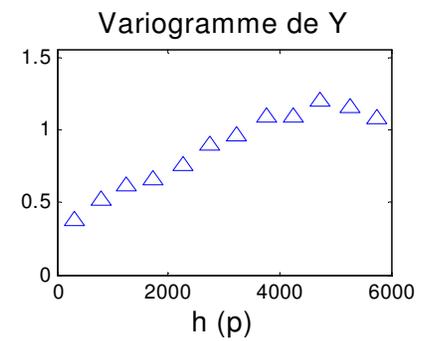
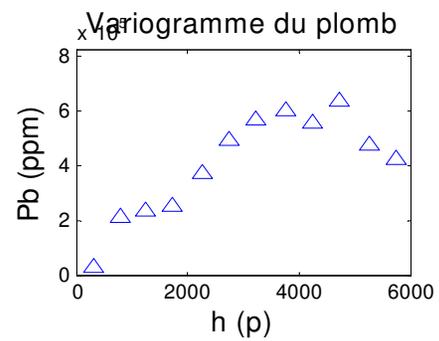
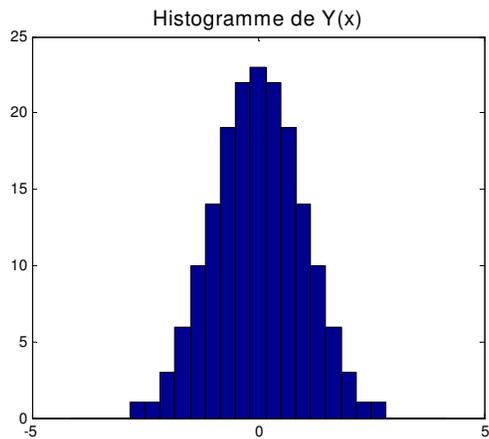
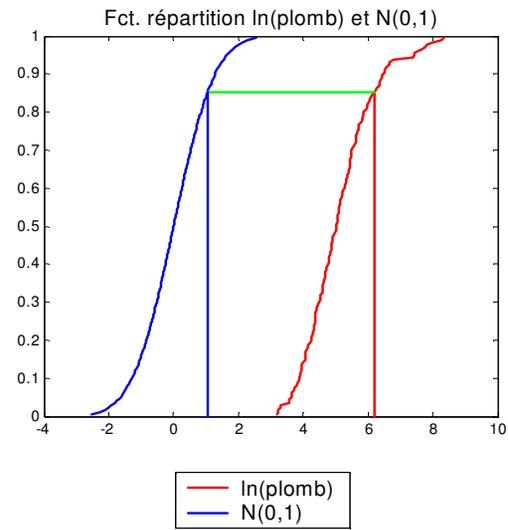
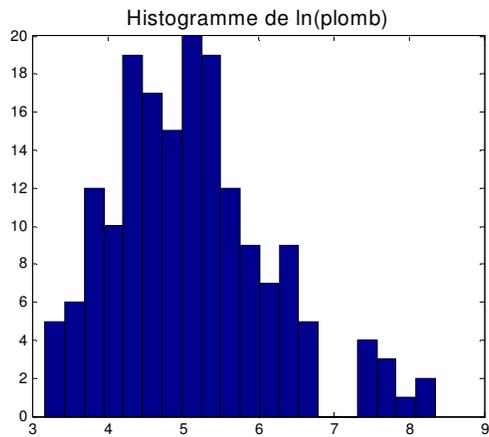
iii. Simuler (conditionnel ou non) $Y(x)$

iv. Effectuer la transformation inverse $Z(x)=f(Y(x))$

En général les
variogrammes de la
variable transformée sont
plus faciles à modéliser

Cette procédure assure la reproduction du variogramme de $Y(x)$. Celui de $Z(x)$ est approximativement reproduit seulement car Y gaussien n'implique pas que $Y(x)$ soit (multi)gaussien,...

Exemple : données de contamination au plomb de Dallas



Simulation de teneurs de blocs

Lorsque la fonction de transfert porte sur des teneurs de blocs (ex. optimisation d'une fosse à ciel ouvert) il faut simuler ces valeurs conditionnellement aux valeurs ponctuelles.

La méthode consiste à simuler des teneurs ponctuelles sur une grille fine, puis de les regrouper selon les blocs désirés (les blocs peuvent être de tailles et de formes différentes).

Recuit simulé

Méthode itérative basée sur l'optimisation d'une fonction objectif

$$\text{Ex. : } O = \underbrace{a^* |\gamma(h) - \gamma_s(h)|}_{\text{1er objectif}} + \underbrace{b^* |f - f_s|}_{\text{2e objectif}} + \dots$$

1er objectif: minimiser
l'écart entre variogramme
théorique et variogramme
simulé

2e objectif: minimiser
l'écart entre la distribution
théorique et la distribution
simulée

3e objectif: ...

La formulation est générale, elle peut accommoder autant d'objectifs différents que désirés pourvu que ces objectifs soient compatibles entre eux.

Algorithme :

- Générer un état initial
- Initialiser la fonction objectif O_0

À une itération « k » donnée

- Sélectionner un point au hasard (disons x_i)
- Tirer une valeur « candidate » d'une distribution « mère » (habituellement la distribution que l'on veut reproduire)
- Calculer la valeur de la fonction objectif obtenue en substituant la valeur candidate à la valeur actuelle au point x_i
- Si la fonction objectif a décru => conserver la valeur candidate.
Si la fonction objectif a augmenté conserver la valeur candidate avec probabilité $p = \exp\{(O_k - O_{\text{candidat}}) / t\}$
et donc conserver la valeur actuelle avec probabilité $(1-p)$
- Si la valeur de la fonction objectif est suffisamment faible ou si on a atteint un nombre maximum d'itérations on arrête, sinon, on retourne à i.

Algorithme recuit simulé (suite)

Note : au point iv, « t » est une *température* que l'on contrôle au moyen d'une *cédule de refroidissement*.

Plus « t » est élevé, plus grande est la probabilité d'accepter une perturbation défavorable.

On amorce l'algorithme avec « t » élevé puis on l'abaisse au fur et à mesure que l'algorithme progresse. Vers la fin, seules les perturbations favorables sont acceptées.

Ex. : $O_k=100$; $O_{\text{candidat}}=110$

=> La perturbation est défavorable

si $t=100 \Rightarrow p=\exp(-10/100) = 0.90$

si $t= 2 \Rightarrow p=\exp(-10/2) = 0.007$

Influence de la cédule de refroidissement

Refroidir trop vite => on risque de se retrouver dans un optimum local dont on ne peut plus s'échapper (i.e. les objectifs ne sont pas atteints)

Refroidir trop lentement => le système oscille beaucoup et la convergence est très lente.

On a peu de guides sur le choix de cette cédule, il faut souvent y aller par essais et erreurs => le recuit simulé demande un certain pif!

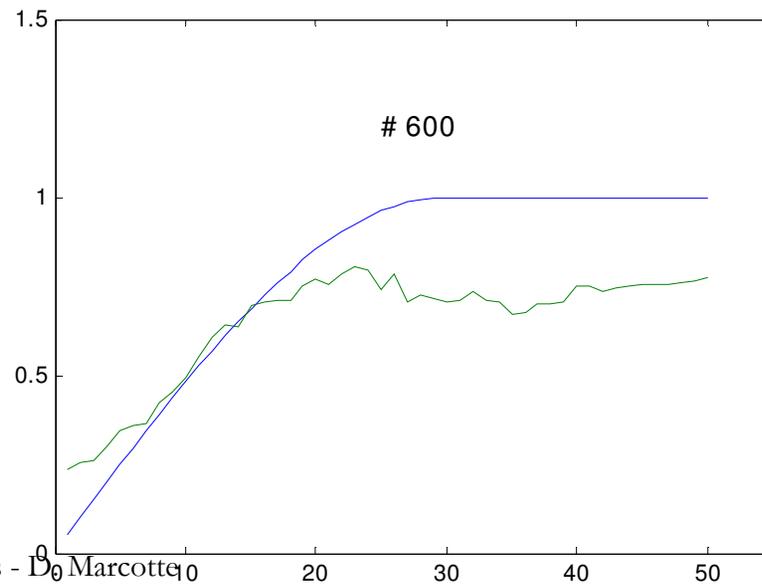
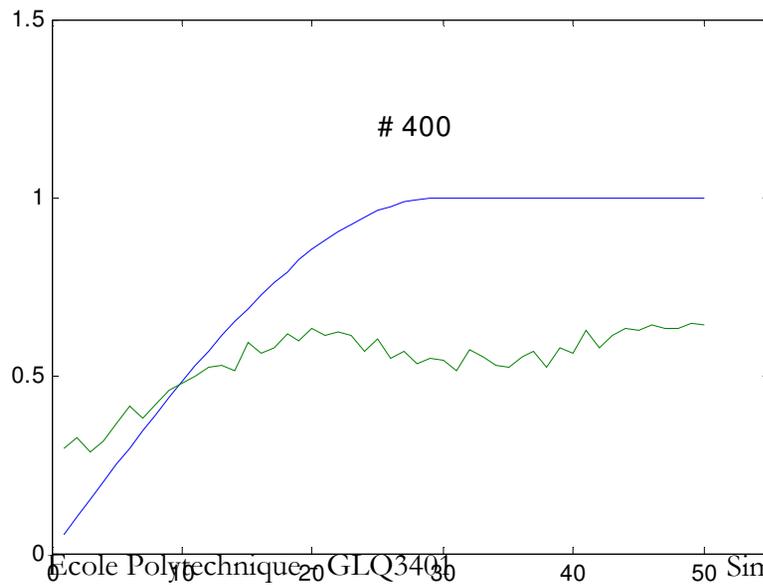
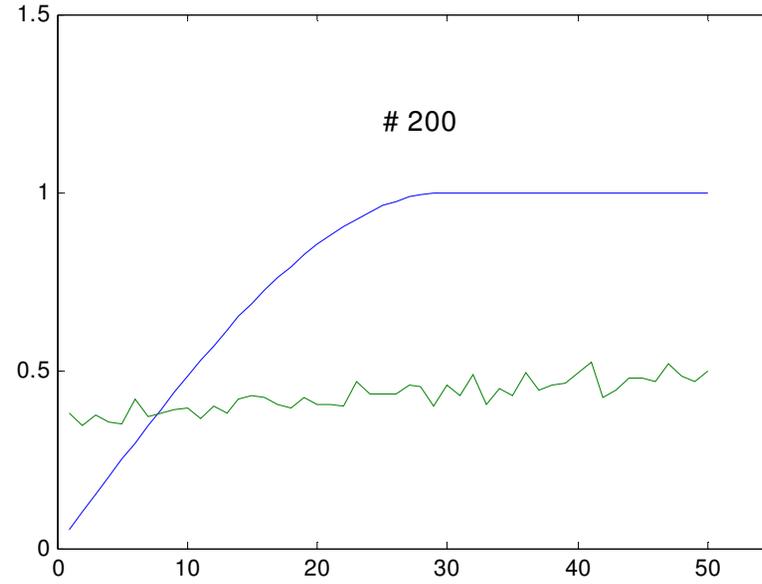
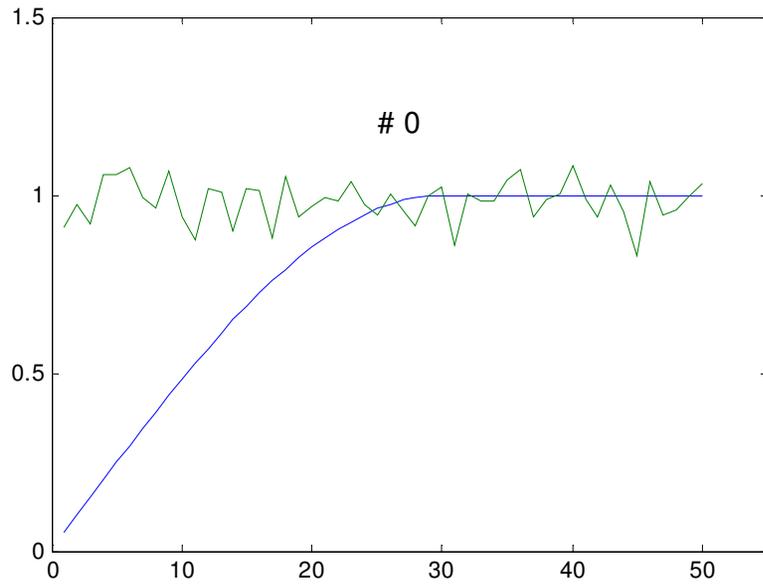
Avantages du recuit simulé

- Facile à comprendre et à programmer
- Très général, tout objectif peut être traité et donc, théoriquement, toute fonction de transfert
- Cas conditionnel immédiat : on n'a qu'à ne jamais perturber un point coïncidant avec une donnée
- Permet de traiter des données non-gaussiennes

Désavantages du recuit simulé

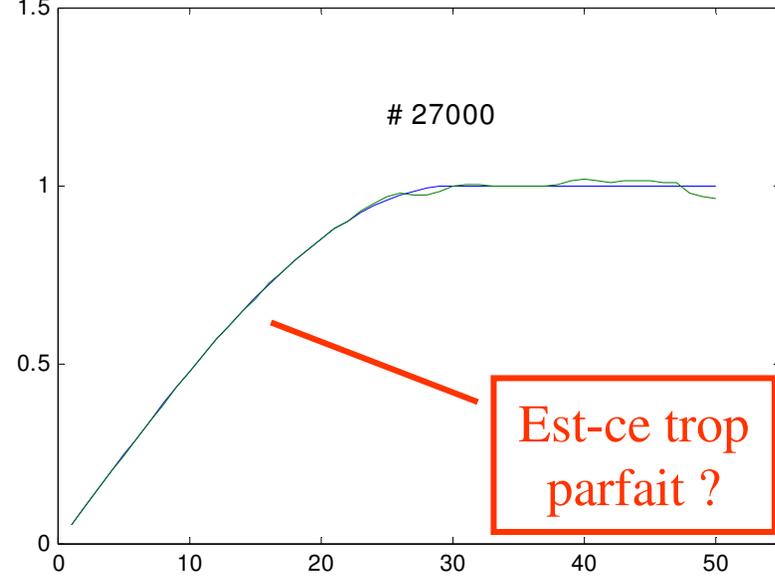
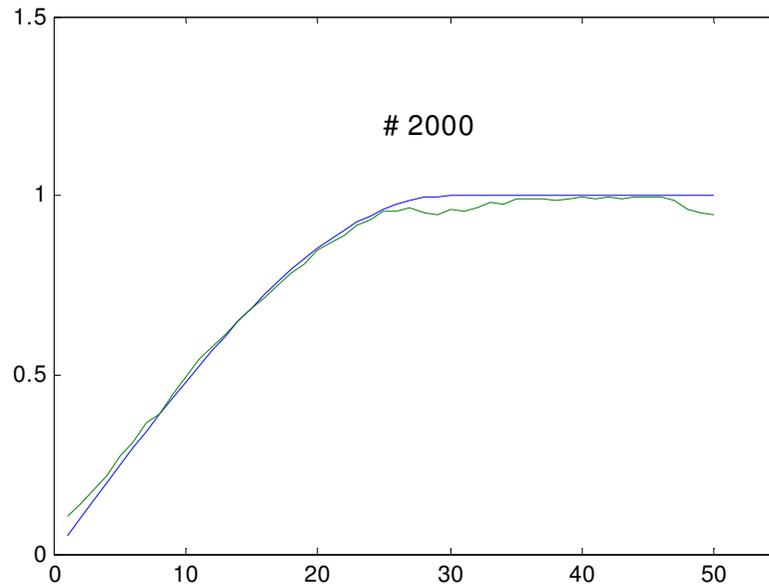
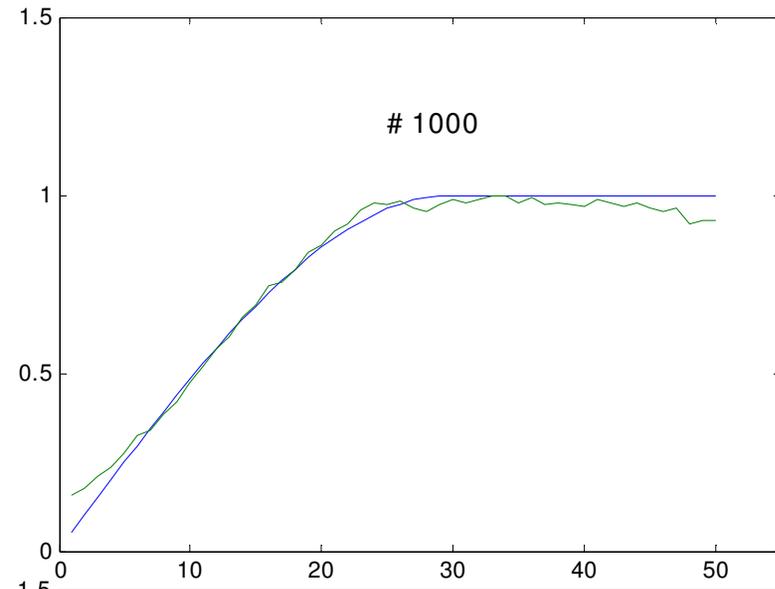
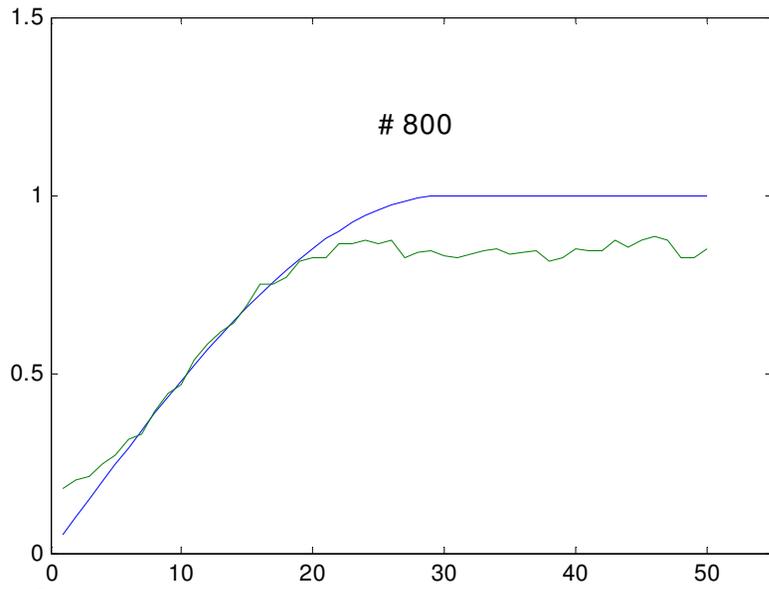
- Difficile de fixer la cédule
- Lent car un seul point est modifié à la fois
- Certains objectifs peuvent être incompatibles sans qu'on le sache
- Si la fonction objectif est longue à calculer, la méthode devient impraticable
- On connaît mal les propriétés statistiques de ce qui est simulé

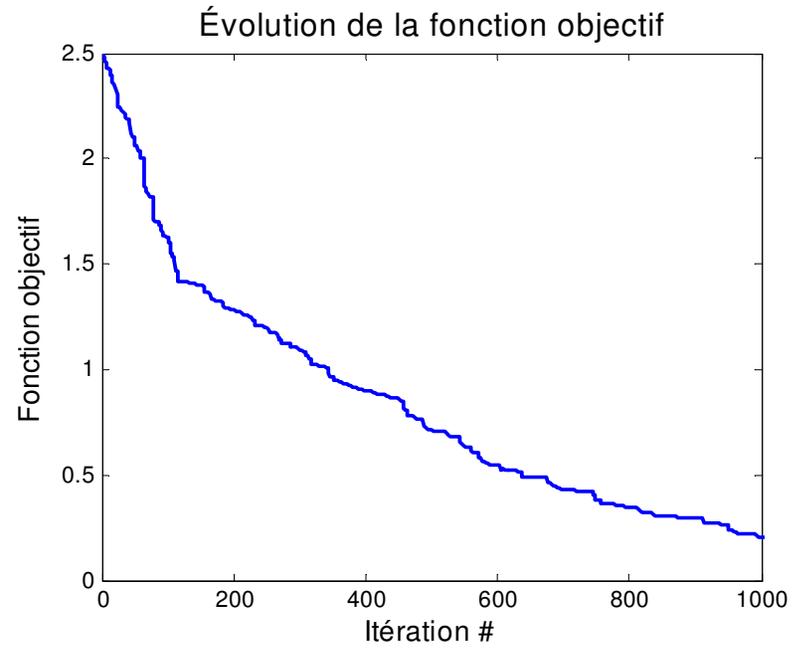
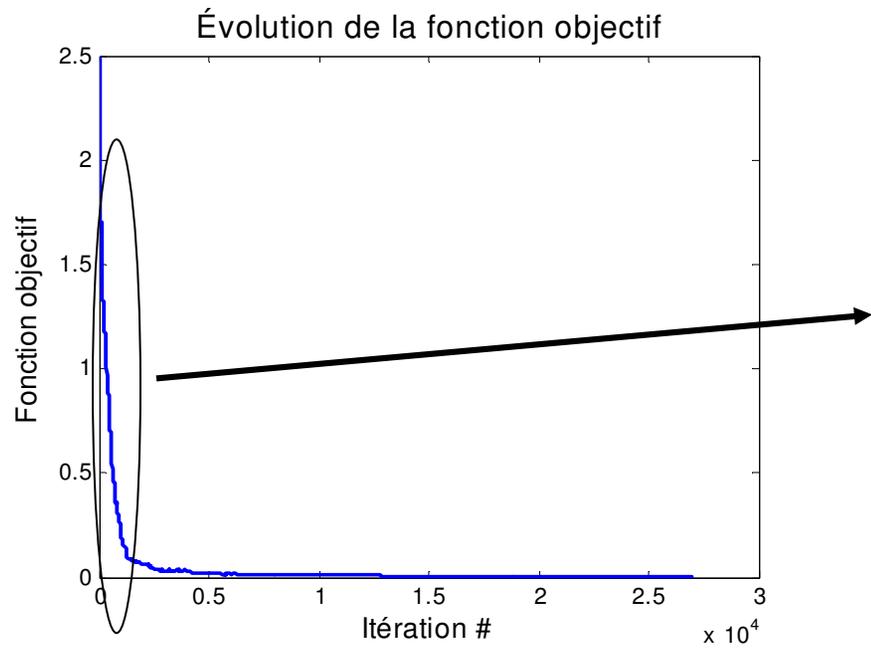
Exemple



0 10 20 30 40 50 Simulations - D_0 Marcotte

Exemple

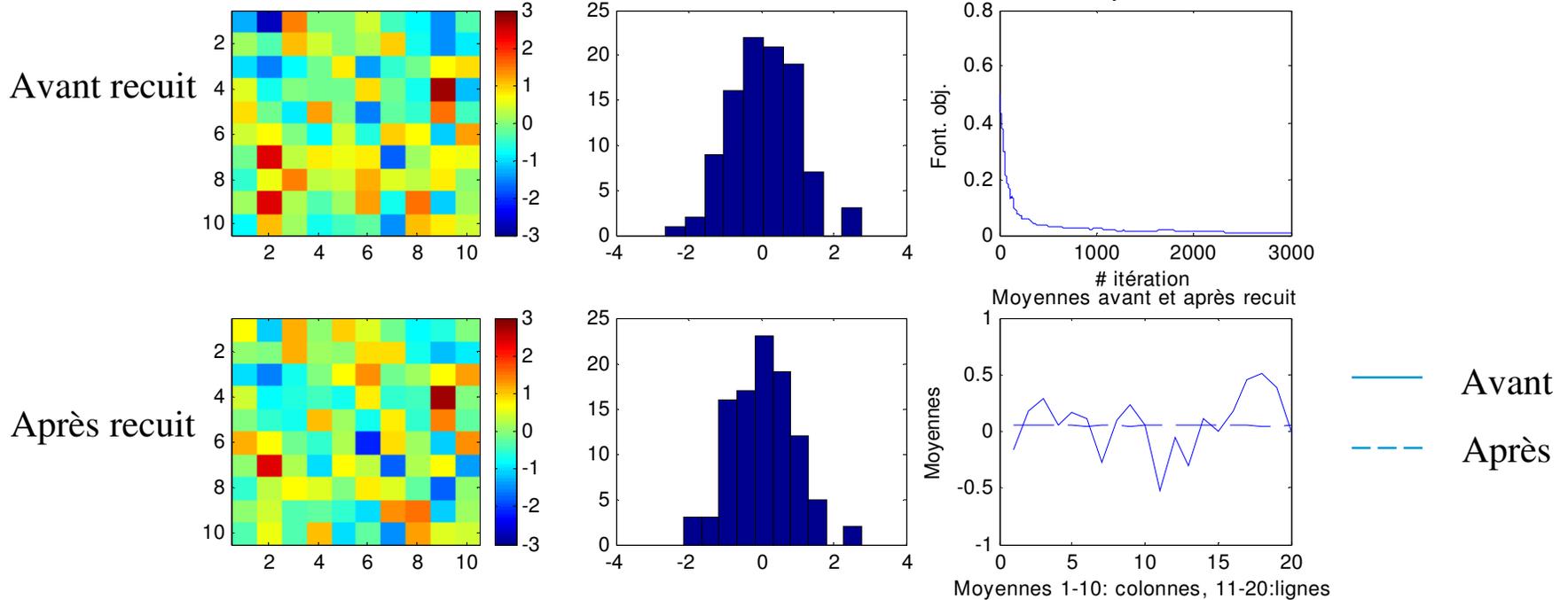




Exemple

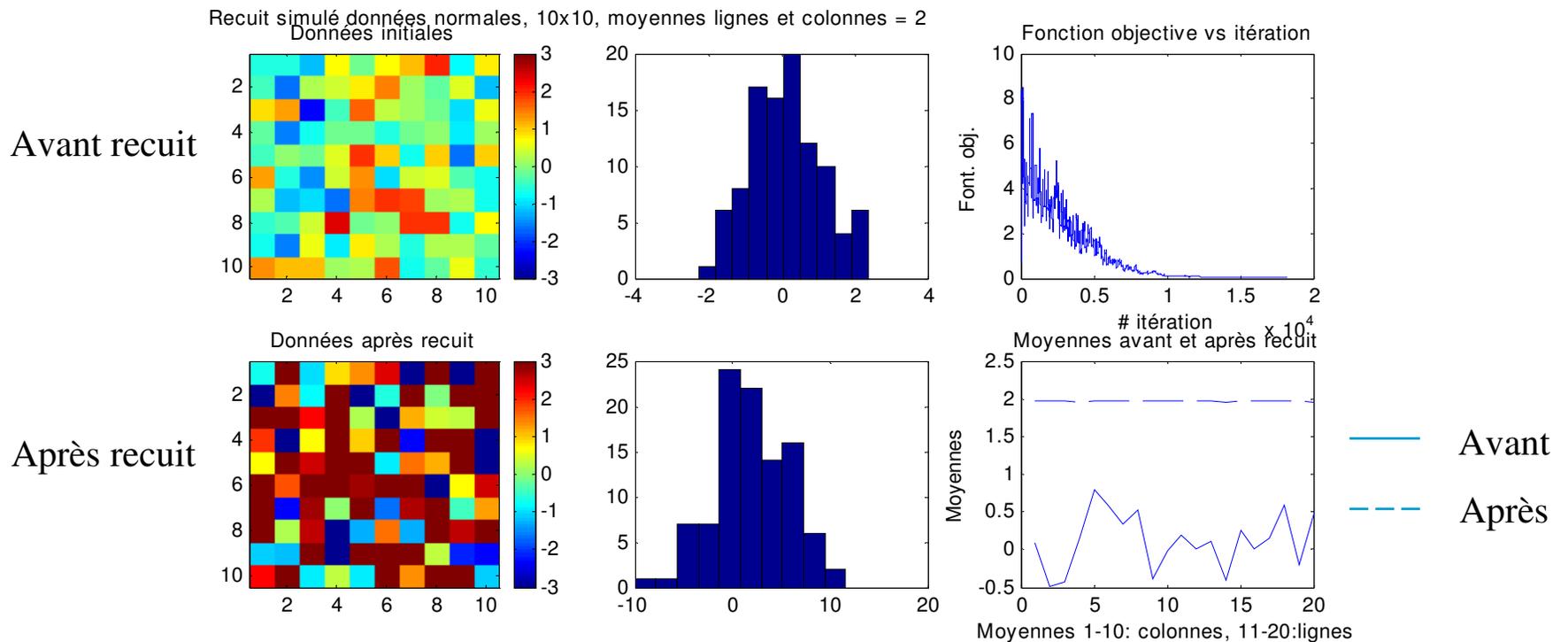
Placer des valeurs dans une matrice de façon à assurer une somme constante en lignes et en colonnes

Recuit simulé données normales, 10x10, moyennes lignes et colonnes égales



Exemple

Placer des valeurs dans une matrice de façon à assurer une moyenne = 2 en lignes et en colonnes. Tirage d'une $N(0,25)$



Post-conditionnement

Des algorithmes plus efficaces que LU, SGS et recuit simulé existent pour fournir des simulations non-conditionnelles

Méthode	Complexité (non-cond.)	Complexité de réalisations additionnelles	Champ
LU	n^3	n^2	Petit
SGS	$m^3 n$	$m^3 n$	Grand à très grand
FFT-ma	$n \log(n)$	$n \log(n)$	Grand
Bandes tournantes	n	n	Très grand

n : nombre de points à simuler

m : nombre de points utilisés dans le krigeage

On peut conditionner des simulations non-conditionnelles à l'aide d'un krigeage simple :

Z_g^s : valeur simulée non conditionnelle à un point de la grille de simulation

Z_i^s : valeur simulée non conditionnelle à un point observation

Z_g^{sc} : valeur simulée à un point de grille conditionnellement aux données

Z_g^{s*} : valeur krigée à un point de grille utilisant les valeurs simulées aux points observation

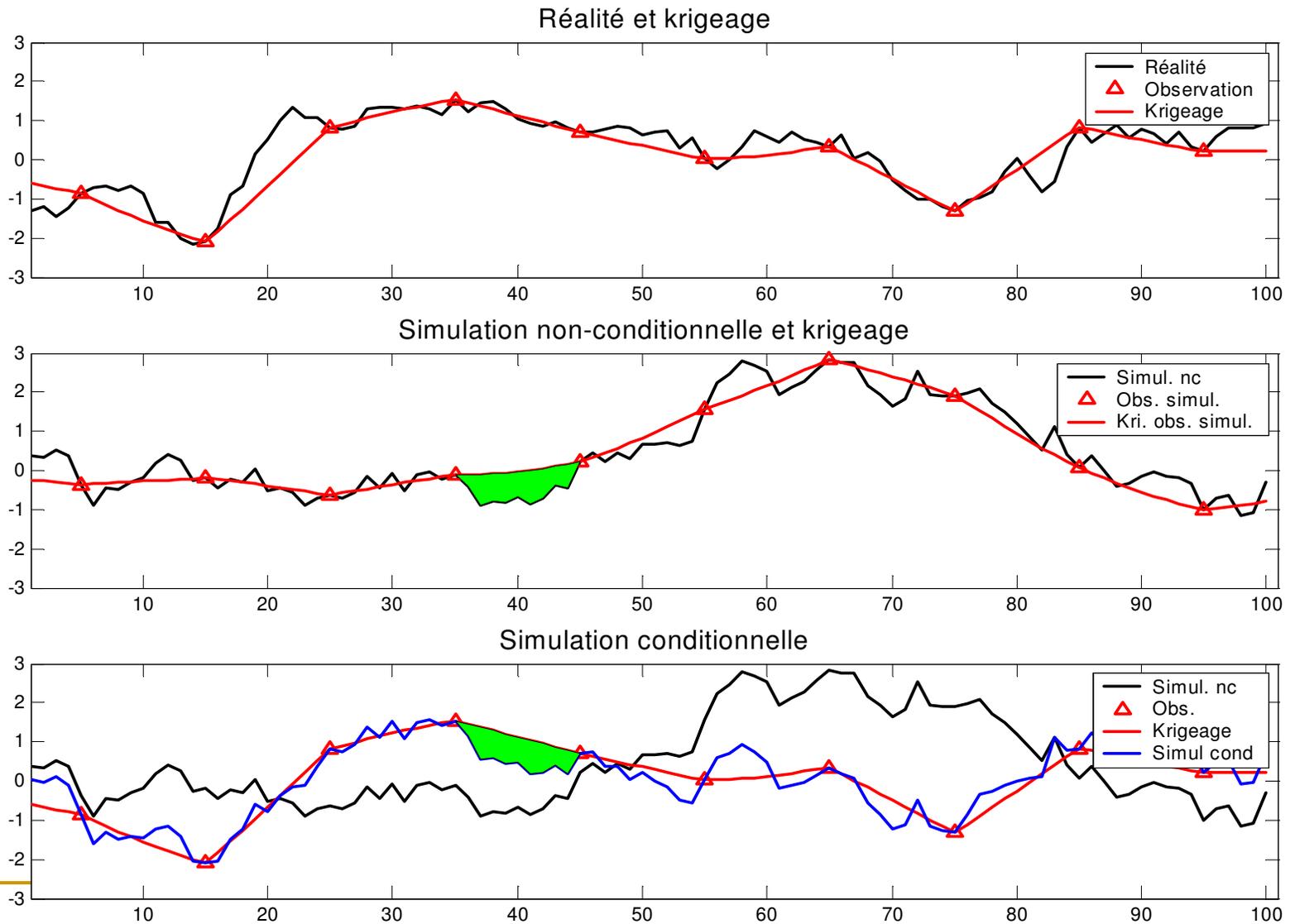
Z_g^* : valeur krigée à un point de grille utilisant les valeurs observées

Z_g : valeur vraie à un point de grille

$$Z_g^{sc} = Z_g^* + (Z_g^s - Z_g^{s*}) \quad \left. \vphantom{Z_g^{sc}} \right\} \text{La valeur krigée + erreur simulée}$$

$$Z_g^{sc} = Z_g^s + (Z_g^* - Z_g^{s*}) \quad \left. \vphantom{Z_g^{sc}} \right\} \text{La valeur simulée + krigeage de la différence aux points observations}$$

Exemple



Propriétés

Non-conditionnelle	Conditionnelle
$E[Z_g^s] = m$	$E[Z_g^{sc}] = Z_g^*$
$\text{Var}(Z_g^s) = \sigma^2$	$\text{Var}(Z_g^{sc}) = \sigma_{ks}^2$
$\text{Var}(Z_g - Z_g^s) = 2\sigma^2$	$\text{Var}(Z_g - Z_g^{sc}) = 2\sigma_{ks}^2$



Pouvez-vous démontrer ces propriétés utilisant la définition du post-conditionnement ?