

# Métamodèles fonctionnels pour Codes de Calcul Physiques



B. Auder

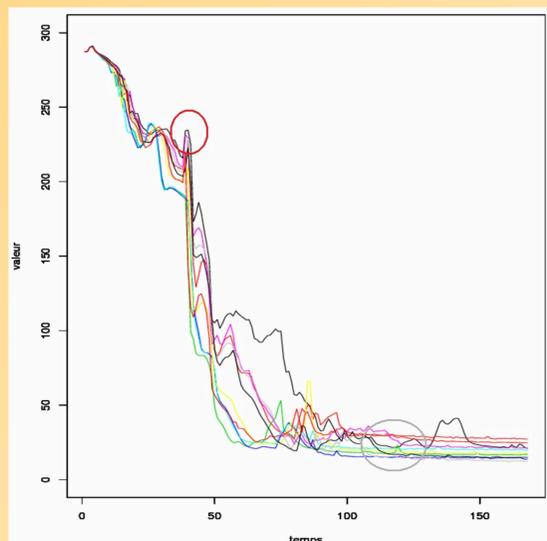
CEA/DEN/CAD/DER/SESI/LCFR -- UPMC Paris 6



## CODE DE CALCUL A ENTREES SCALAIRES ET SORTIES FONCTIONNELLES

$$\begin{pmatrix} x_1 = (x_{11}, \dots, x_{1p}) \\ \vdots \\ x_i = (x_{i1}, \dots, x_{ip}) \\ \vdots \\ x_N = (x_{N1}, \dots, x_{Np}) \end{pmatrix}$$

$$\begin{pmatrix} y_1(t) \\ \vdots \\ y_i(t) \\ \vdots \\ y_N(t) \end{pmatrix}$$



Caractéristiques (très) intéressantes [ovale rouge]  
..et probablement moins vitales [ovale gris]

Comment les déterminer / classer automatiquement ?

→ Avis d'expert, et / ou :  
→ Tests statistiques, lissage, leave-one-out ..etc

## PROPAGATION DES INCERTITUDES

Incertitudes sur les paramètres (entrées du code)

Incertitudes sur la sortie du code de calcul

### Problématiques :

■ Quels sont les paramètres les plus influents ? (C.-à-d. ceux dont la variabilité entraîne les plus fortes oscillations du résultat)

#### ANALYSE DE SENSIBILITE

■ Comment la variabilité des entrées se répercute sur la sortie du code de calcul ? Incertitude au final sur la sortie ?

Estimation de la marge de confiance sur la prise de décision ?

#### PROPAGATION D'INCERTITUDES

→ Analyses difficiles si code **f** trop lent, donc :  
Construction d'un **métamodèle**  $\Phi$  (validé par leave-one-out par exemple, le plus fidèle possible) :

$$\phi : x \in \mathbb{R}^p \mapsto y_x(t) \in \mathcal{F}([a, b], \mathbb{R}),$$

Supposant hypothèses de régularité sur  $f$ , et :

$$x_1, \dots, x_N ; y_{x_1} = f(x_1), \dots, y_{x_N} = f(x_N) \text{ connus}$$

## CONSTRUCTION DU METAMODELE EN 3 PHASES

### I) Classement des courbes en groupes de similarité

→ - Facilite l'apprentissage (réduction de la complexité)  
- Améliore l'interprétation du modèle final

Inconvénient : nécessite un classement (supervisé) des entrées correspondantes avec marge d'erreur associée

### II) Réduction de la dimension :

- Approximation des fonctions par projection sur une base  
→ Double effet : lissage, et passage dans  $\mathbb{R}^d$   
→ Donc très intéressant si courbes bruitées

Ou :  
- Simplification directe des fonctions sur certaines zones pré-déterminées, au second plan dans les études de courbes.

Ou rien, auquel cas : métamodèle long à construire, corrélations inter-courbes difficiles à prendre en compte.

### III) Reconstitution, prédiction

Apprentissage statistique sur les coefficients de décomposition :  
Étape classique avec de nombreuses solutions proposées.

#### Etape I : Définition d'une distance pertinente ?

Clustering spectral assez efficace ; alternatives :  
k-Means, algorithm E-M, clustering hiérarchique ..etc

#### Etape II : Comment relier le pouvoir explicatif d'une base au taux de mauvais classement attendu ?

Bases : ACP, splines, ondelettes ..etc

#### Etape III : SVM, processus gaussiens, réseaux de neurones, bases de splines, boosting ..etc (ou ..)

### Conclusion :

Beaucoup de travail à faire sur les points 1 et 2

→ Nombre optimal de clusters ?

→ Impact d'une erreur de classification ?

→ Base de fonction optimale ?

### En cours :

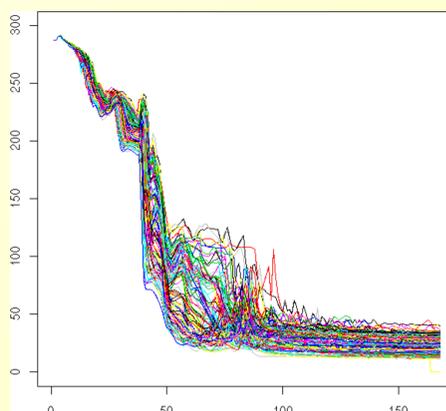
- tests d'heuristiques pour le nombre de clusters

- tentative de généraliser l'ACP fonctionnelle

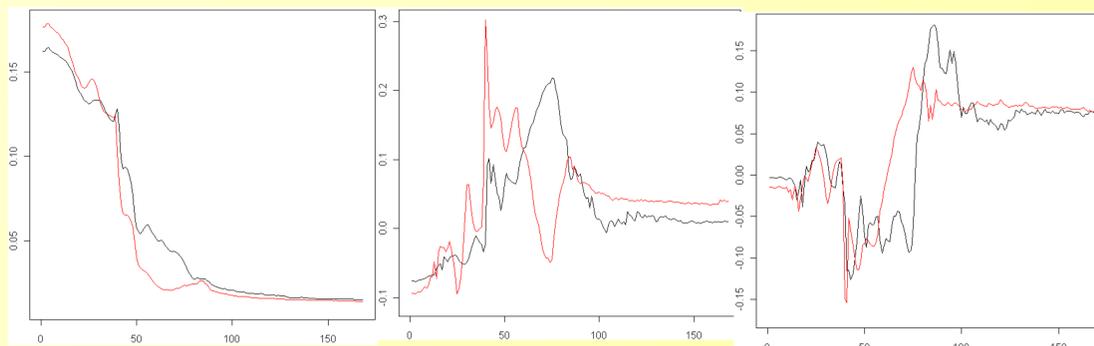
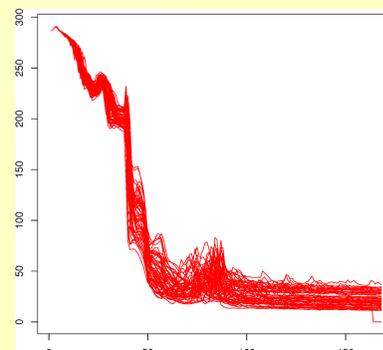
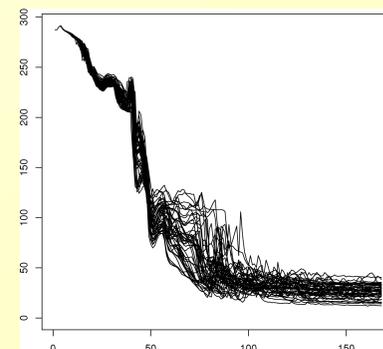
$p = 4, d = 3$  ; 100 courbes dont 48 dans un cluster, 52 dans l'autre.

Cas général :

Comment déterminer le nombre de groupes ?

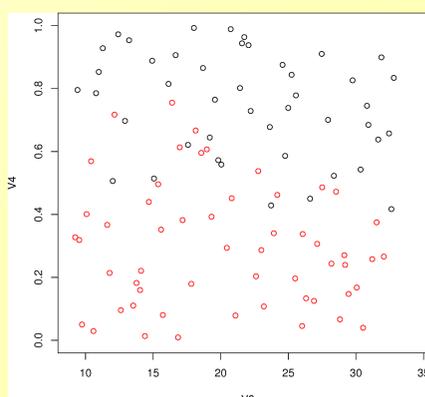


Premières fonctions de base ACP :  
(+90% de variabilité expliquée)

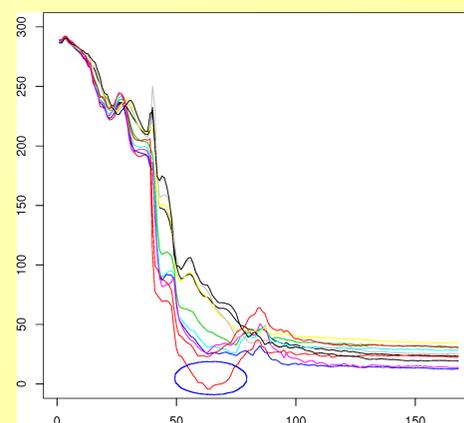


Scatter-plot entrées :

Composante 3 vs. 4



Quelques courbes prédites :



→ Arbre de décision bien adapté, taux d'erreur < 10% . Suffisant ?

Zone entourée en bleu : probablement une mauvaise classification de l'entrée à prédire.