

Simulateurs multifidélités : quelques questions issues de problèmes industriels

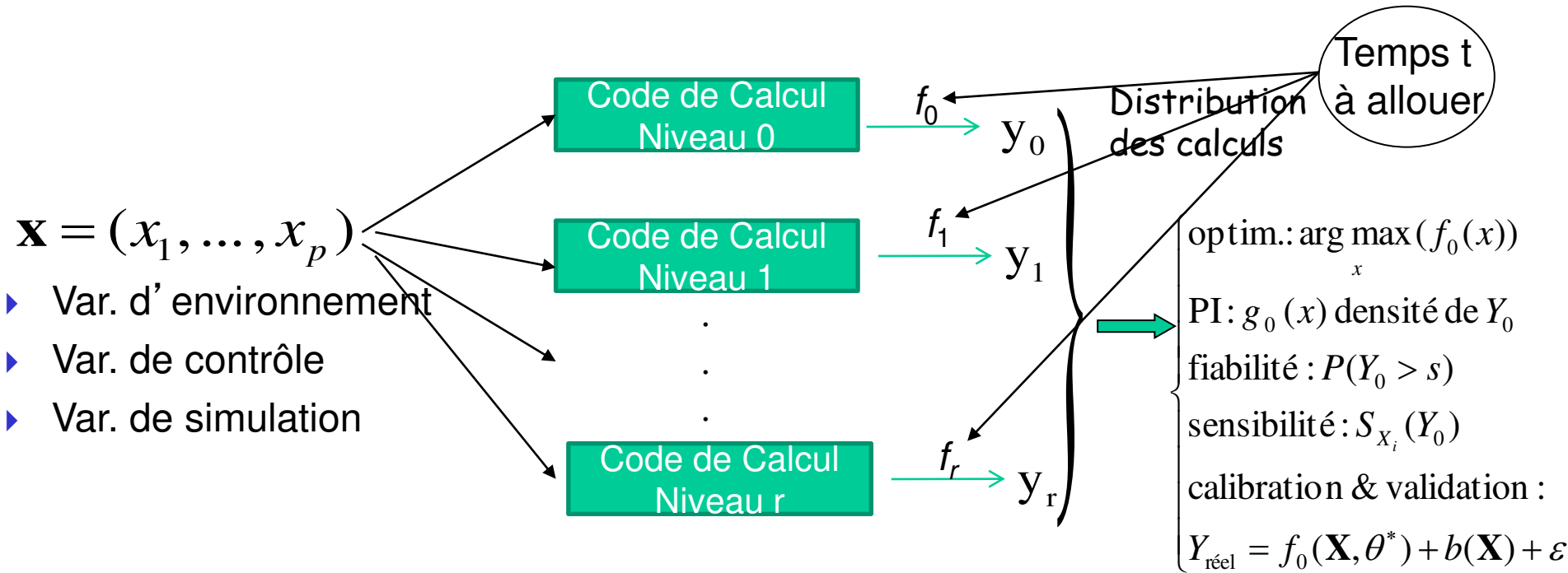
Bertrand Iooss

EDF R&D
Département Management
des Risques Industriels



Objectif général : Exploiter la possibilité de certains simulateurs d'être dégradés (avec un tps cpu diminué), pour réduire les ressources en temps de calcul nécessaires à une étude d'incertitudes (calcul de fiabilité, analyse de sensibilité, propagation d'incertitudes, calibration, ...)

Vaincre le coût cpu : simulateurs multi-fidélités



Remarques :

- Les différents niveaux de fidélité sont prédéfinis par le modélisateur
- Certains problèmes ont été largement explorés (ex : optimisation), d'autres effleurés (validation) et d'autres non traités

Objectif : proposer des stratégies optimales d'allocation de calculs

Des solutions à base de krigeage/cokrigeage que l'on ne rappelle pas ici (cf. ce matin)

Tentative de classification des simulateurs multifidélités

- Codes à convergence numérique dégradée :
 - Maillage éléments finis (ou différences finies, etc.) – exemple : propagation d'ondes élastiques, mécanique du solide
 - Non convergence/convergence partielle – exemple : écoulements (CFD)
 - Codes Monte Carlo – exemple : neutronique, simul. lagrangienne
- La réponse d'un calcul est une moyenne (ou autre opérateur stat.) de n simulations aléatoires

- Simulateurs physico-numérique (phénoménologique, multi-D) vs. modèles de « décision » (modèle d'action, modèle réduit, modèle opérationnel, abaque, outil temps réel, modèle 0D ou 1D)

Questions :

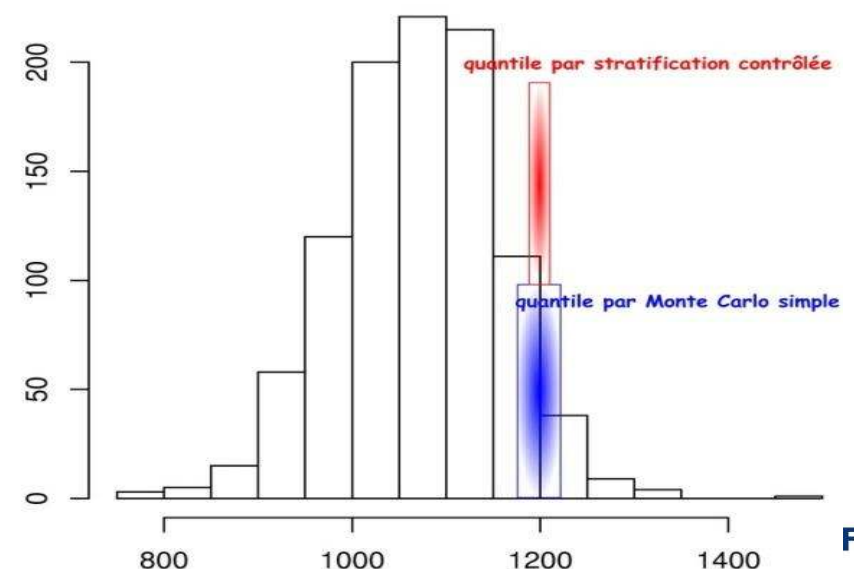
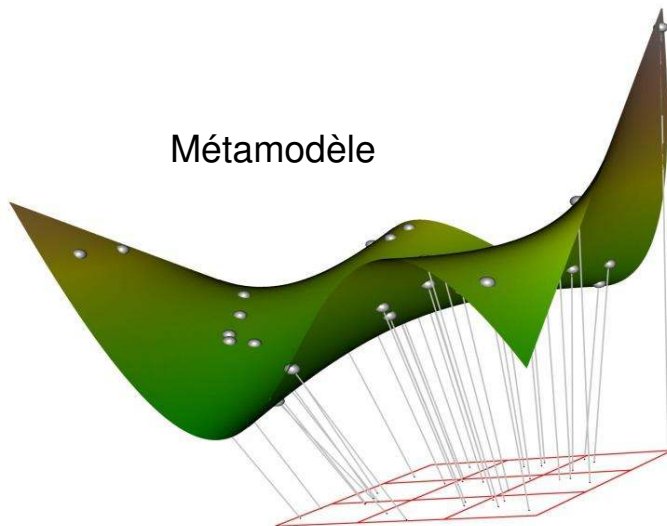
- 1) Développer un modèle de décision à partir d'un modèle de connaissance ?
- 2) Enrichir un modèle de décision à partir d'un modèle de connaissance ?
- 3) Exploiter de manière adaptative les deux types de modèles

- Métamodèles + ou – fins : régression polynomiale, krigeage basique, polynome de chaos, etc...

Exemple : estimer le quantile d'une sortie d'un code de calcul

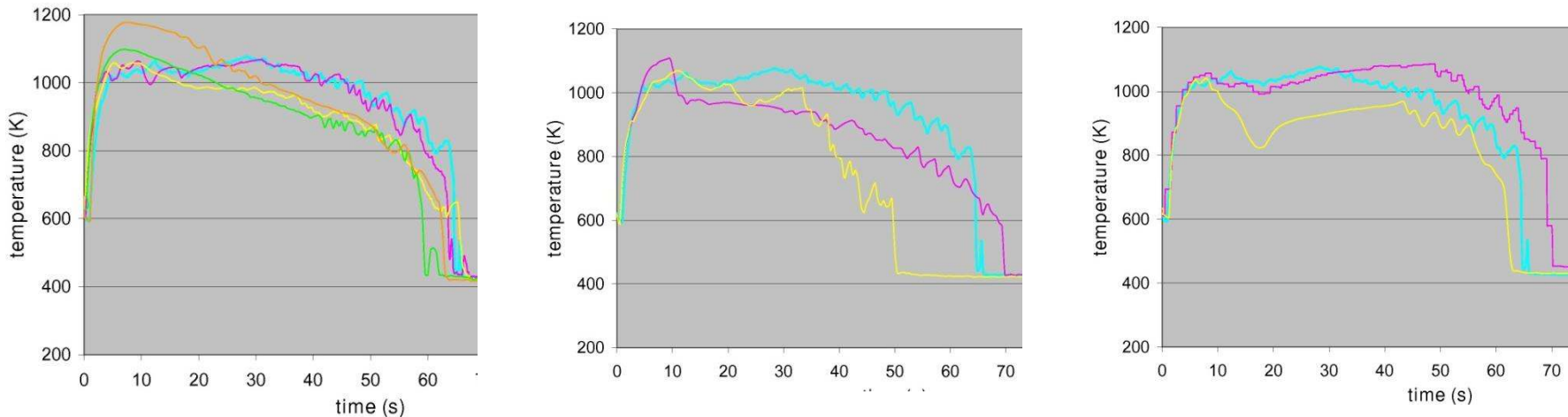
Trouver $\hat{Y}(\alpha, n)$ estimation de y_α / $P(Y \leq y_\alpha) = \alpha$ à partir de (Y_1, \dots, Y_n)
 $Y = f_0(\mathbf{X})$ est la sortie d'un code de calcul, $\mathbf{X} = (X_1, \dots, X_d)$ vecteur aléat.

- Méthode classique basée sur l'estimateur empirique (Monte Carlo simple)
Mais variance élevée
 - Couplage judicieux entre approches stats (apprentissage) et proba (Monte Carlo) : *utiliser un métamodèle $f_r(\mathbf{X})$ pour trouver des échantillons de \mathbf{X} dans des zones intéressantes, sur lesquelles $f_0(\cdot)$ est calculée*
- Stratification, contrôlée, tirage d'importance contrôlée [Cannamela et al. 2009]



Autre type de problème : simulateurs « concurrentiels »

Exemple : benchmark BEMUSE en thermohydraulique accidentelle des REPs



- Prédiction par 8 équipes différentes et réponses expérimentales (en bleu)
- Différences de codes, phénomènes modélisés parfois différents, mises en données différentes, différents choix entre options numériques de résolution et/ou types de conditions aux limites, effets « utilisateurs », ...

On peut songer à agréger les codes, extraire le meilleur code, tirer partie du meilleur des codes dans différentes zones des entrées, utiliser le « pire » des codes lors d'une évaluation de risque

=> Questions de validation

Quelques questions difficiles

- Le niveau de fidélité est souvent variable en fonction de \mathbf{x} , il faut aussi une étape d'apprentissage de l'erreur entre modèles
- Le temps de calcul de chaque modèle peut dépendre de \mathbf{x}
- Cas d'un continuum de niveaux de fidélité
- Stratégies adaptatives pour sorties multiples ?
- Cas avec parallélisation (lancement de grappes de calculs) ?
- Simulateurs ayant certaines de leurs entrées différentes
- Plans d'expériences initiaux : imbriqués/non imbriqués, ayant certaines entrées différentes, etc.
- ...