Journée Utilisateurs #16 OpenTURNS

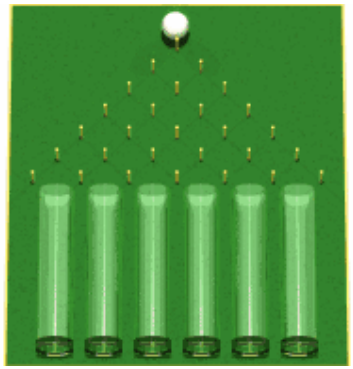
Module oteclm
Extended Common Load Method

V. Rychkov – A. Dutfoy

EDF R&D Pericles

Vendredi 23 juin 2023

EDF Lab Saclay



Introduction

❖ **Sûreté nucléaire à EDF**

- Dès la construction du parc nucléaire (fin 1970)
- Etudes déterministes / **probabilistes** (**EPS**) (depuis 1991)
- Les EPS sont organisées par **séquences accidentelles** qui contiennent des événements initiateurs.
- L'objectif des EPS est de calculer le **risque de fusion du coeur** (EPS de niveau 1).

❖ **Systèmes importants pour la sûreté**

- Conçus avec une redondance élevée pour assurer un **haut niveau de fiabilité**
- Les **défaillances de cause commune** (**DCC**) sont particulièrement étudiées car elles sont les **principales responsables de défaillance des systèmes redondants**.
- Certains systèmes sont composés **de plusieurs dizaines de composants** : par exemple, les grappes de contrôle, les soupapes de sûreté (REB).
- Dans ce cas, on parle de **défaillance de cause commune de grand ordre**.

Introduction

❖ **Fonctions de sûreté**

- Contrôle de réactivité
- Refroidissement
- Confinement

❖ **Contrôle de réactivité: Arrêt Automatique du Réacteur (AAR) - Grappes de contrôle**

- En cas d'un aléa (perturbation important de fonctionnement de la centrale) **l'Arrêt automatique du réacteur est automatiquement enclenché.**
- Le système d'arrêt d'urgence du réacteur agit pour limiter les conséquences des conditions de fonctionnement, **c'est-a-dire limiter rapidement l'énergie dégagée** par le coeur du réacteur et **faciliter le refroidissement du combustible .**
- Les **grappes de contrôle** constituées de crayons absorbant les neutrons et sont insérés dans les assemblages combustibles. (60-90 grappes de contrôle)
- Dans les Etudes Probabilistes de Sûreté, **l'AAR est défaillant** lorsqu'il y a non insertion de N grappes (N=1,2, 3, … selon le scenario).

Introduction

❖ **Plusieurs méthodes pour quantifier les défaillances de cause commune**

- Modèle du facteur β de Fleming (1975)
- Modèle Multiple Greek Letter de Flemming
- Modèle du facteur α de Molesh et Siu (1987)
- ➔ modèles peu efficaces pour un grand nombre de composants redondants!

❖ **Extended Common Load Model**

- Méthodologie pour **les DCC de grand ordre**.
- Avantage: le nombre de paramètres est **indépendant du nombre de composants**.
- Développée par **T. Mankamo** dans les années 90.
- Modèle utilisé par EDF pour le calcul des probabilités de défaillances simultanées de k composants parmi n, pour les grappes de contrôle.

ECLM

❖ **Système étudié:**

- n composants **identiques**,
- de **même résistance R**
- soumis à une **même sollicitation S**

Sollicitation aléatoire, modélisée par une mixture de lois Normale

$$f_S(s) = \pi \frac{1}{\sigma_b} \varphi\left(\frac{s-\mu_b}{\sigma_b}\right) + (1-\pi)\frac{1}{\sigma_x}\varphi\left(\frac{s-\mu_x}{\sigma_x}\right) \quad \forall s \in \mathbb{R}$$

Résistance aléatoire, modélisée par une loi Normale

$$f_R(r) = \frac{1}{\sigma_R}\varphi\left(\frac{r-\mu_R}{\sigma_R}\right) \quad \forall r \in \mathbb{R}$$

❖ **Probabilités de l'ECLM:**

- **PSG(k|n)**: prob. que k composants spécifiques soient défaillants (quel que soit l'état des autres composants)
- **PEG(k|n)**: prob. que k composants spécifiques soient défaillants et que les autres soient fonctionnels
- ***PES(k|n)***: prob. que k composants quelconques soient défaillants et que les autres soient fonctionnels
- **PTS(k|n)**: prob. que k composants quelconques soient défaillants (quel que soit l'état des autres composants)

ECLM

- La **connaissance des PEG(k|n)** permet de connaître les PSG(k|n), les PES(k|n) et les PTS(k|n)
- On se focalise sur les probabilités PEG

$$\mathrm{PEG}(k|n) = \mathbb{P}\left[S > R_1, \ldots, S > R_k, S < R_{k+1}, \ldots, S < R_n\right]$$

$$\mathrm{PEG}(k|n) = \int_{s\in\mathbb{R}} f_S(s)\left[F_R(s)\right]^k \left[1 - F_R(s)\right]^{n-k} ds$$

➔ estimation des paramètres du modèle à partir des données observées

❖ **Données disponibles:**

- Décompte du nombre de composants en panne parmi n à chaque sollicitation,
- Un grand nombre N de sollicitations
- Vecteur d'impact: $V = (V_0, \ldots, V_n)$ où V_i est le nombre de sollicitations ayant mené à i composants défaillants exactement parmi n

$$V_0 + \ldots + V_n = N$$

ECLM

❖ **Modèle probabiliste:**

- Le nombre de sollicitations parmi N qui ont mené à k composants HS exactement suit une loi **MultiNomiale paramétrée par (N, (p_0, …, p_n))** avec:

$$(p_0, \ldots, p_n) = (\text{PES}(0|n), \ldots, \text{PES}(n|n))$$

❖ ***PES(k|n)***: prob. que k composants quelconques soient défaillants et que les autres soient fonctionnels

- **Le vecteur d'impact est une réalisation de cette loi MultiNomiale**

❖ **Vraisemblance des données:**

- Vecteur des paramètres du modèle: $\boldsymbol{\theta}_t = (\pi, \mu_b, \sigma_b, \mu_x, \sigma_x, \mu_R, \sigma_R)$

$$\mathcal{L}(\boldsymbol{\theta}_t | V_t^{n,N}) = \prod_{k=0}^{n} \text{PES}(k|n)^{V_t^{n,N}[k]}$$

$$\log \mathcal{L}(\boldsymbol{\theta}_t | V_t^{n,N}) = \sum_{k=0}^{n} V_t^{n,N}[k] \log \text{PES}(k|n)$$

❖ Comme PES(k|n) = C_n^k PEG(k|n), on peut écrire la log-vraisemblance en fonction des PEG(k|n) (à une constante près)

$$\boldsymbol{\theta}_t^{optim} = \arg\max_{\boldsymbol{\theta}_t} \sum_{k=0}^{n} V_t^{n,N}[k] \log \text{PES}(k|n) = \arg\max_{\boldsymbol{\theta}_t} \sum_{k=0}^{n} V_t^{n,N}[k] \log \text{PEG}(k|n)$$

ECLM : module oteclm

$$\boldsymbol{\theta}_t^{optim} = \arg\max_{\boldsymbol{\theta}_t} \sum_{k=0}^{n} V_t^{n,N}[k] \log \mathrm{PES}(k|n) = \arg\max_{\boldsymbol{\theta}_t} \sum_{k=0}^{n} V_t^{n,N}[k] \log \mathrm{PEG}(k|n)$$

❖ **Procédure:**

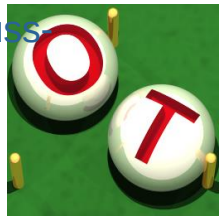
- Nouveau paramétrage du problème
- + Hypothèse de Mankamo

➔ l'optimisation de la log-vraisemblance est une problème de **dimension 3**

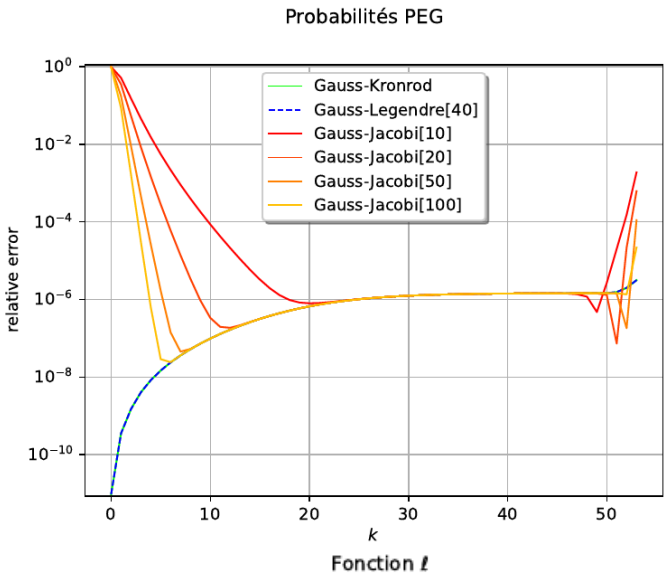
- Bootstrap sur les données pur obtenir plusieurs réalisations du paramétrage optimal et estimer sa loi

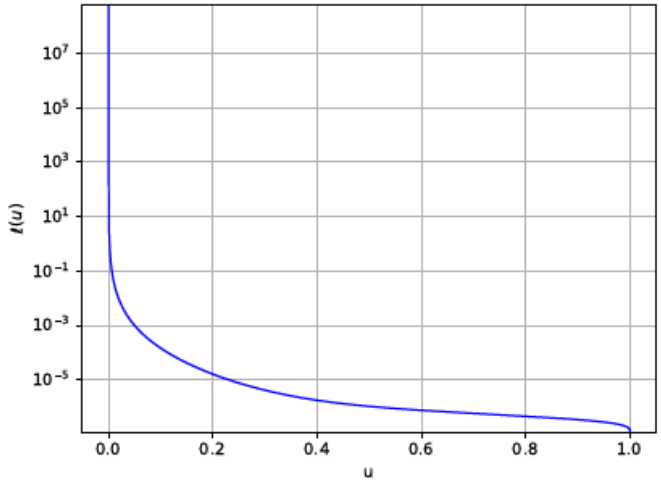
❖ **Diificultés:**

- La maximisation de la vraisemblance nécessite le **calcul de (n+1) intégrales**: PEG(k|n) à chaque appel de la log-vraisemblance
- Ce calcul doit être précis **même pour des probabilités très faibles** car les probabilités interviennent sous forme de log PEG(k|n) dans la vraisemblance!
- Nous avons utilisé les méthodes de quadrature proposées par OpenTURNS: Gauss-Legendre, Gauss-Kronrod
  - ➔ erreur relative inférieure à 10^{-6} pour toutes les PEG (même 10^{-8} pour k >6)
  - ➔ Par défaut, GL(40 points)



ECLM : module oteclm





❖ Quadratures

- Calculs validé /Maple
- Gauss-Jacobi:

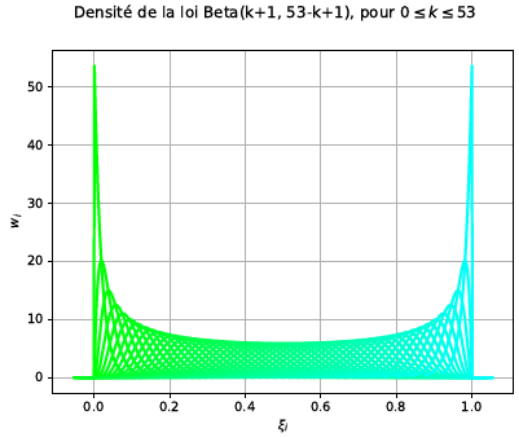
$$\mathrm{PEG}(k|n = \beta(n+1, n-k+1) \int_0^1 \ell(u)\, p_k(u)\, du$$

- La fonction l a une variation très brutale au voisinage de 0: elle passe de 10^7en u=0 à 10^{-5} en u=0,2!
- La décroissance en 0 est beaucoup plus rapide que n'importe quel polynôme

➔ l est bien approchée par un polynôme en u=1 et très mal en u=0

❖ **Pour k grand, les points de la quadrature sont autour de 1 ➔ calcul de bonne qualité!**

❖ **Pour k petit, les points de quadrature sont autour de u=0 ➔ calcul de mauvaise qualité!**



ECLM : module oteclm

❖ **Performance du code:**

- **Optimisation de la log-vraisemblance**: on a choisi **Cobyla** pour ne pas avoir à calculer le gradient de la log-vraisemblance par rapport aux paramètres
- **Fonctions**: nous avons utilisé le plus possible la classe **SymbolicFunction** beaucoup plus performante pour l'évaluation d'une fonction que la classe PythonFunction qui s'appuie sur une fonction définie en python
- **Mécanisme de cache**: nous avons mis en place ce mécanisme pour le calcul des probabilités PEG
- **Parcimonie des calculs**: dans l'étape d'optimisation de la log-vraisemblance, ne sont calculés que les PEG(k|n) associées à une composante V_k non nulle!
- **Parallélisation**: pour les calculs massifs par Bootstrap, nous avons mis en place une technique de parallélisation basée sur l'écriture d'un fichier annexe et qui lance la parallélisation via la bibliothèque **Pool** (utilise tous les coeurs de calcul disponibles sur la machine)

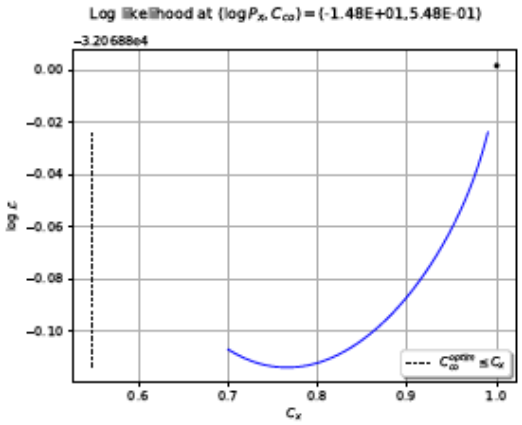
ECLM : module oteclm

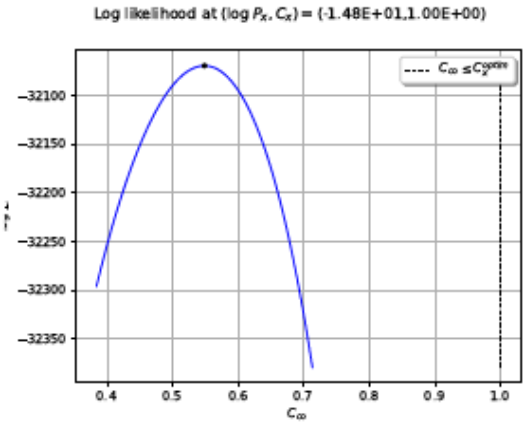
❖ **Exemple:**

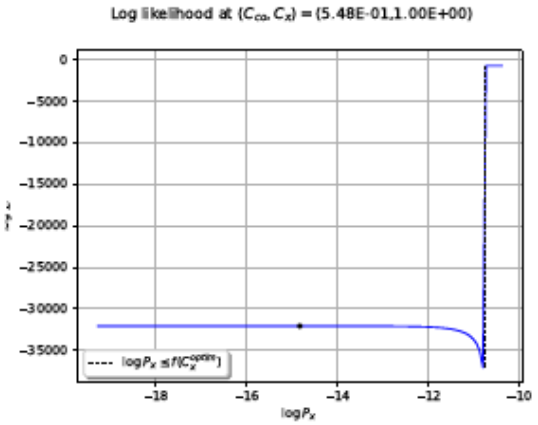
- 53 composants
- Sollicités 2797 fois

Les résultats ont été supprimés

ECLM : module oteclm







Conclusion

❖ **Pourquoi un module OpenTURNS?**

- **Profiter des capacités de modélisation de l'outil**

➔ Etude d'incertitudes possible!

➔ Méthodes numériques pour le calcul des intégrales

- **Code open source**

➔ Diffusion auprès de la communauté EPS

➔ Capitalisation / enrichissement

➔ Audit aisé des autorités de contrôle

➔ Acceptation de la méthode