

Rapport de stage PHIMECA

Kristof Attila Simady
e-mail: kristof.simady@sigma-clermont.fr

05/03/2020

Contents

1	Introduction	2
2	Recherche bibliographique	2
2.1	Travaux retenus	3
2.2	Methodes et outils de travail	3
2.3	Prémices	3
2.3.1	Mission confiée et tâches associées	3
2.3.2	Prise en main d'openTURNS et de la théorie sur les champs stochastiques	3
2.4	Exemple théorique simple, premiers codes	6
2.5	Choix des codes et difficultés algorithmiques	8
2.6	Example of verbatim text	10
2.7	Lists	10
3	Results	10
4	Tables and figures	11
5	Discussion	11
6	Conclusions	11

Avant propos

Le cadre d'étude offert par SIGMA Clermont a donné la possibilité d'effectuer une année supplémentaire de stage pour étoffer nos connaissances scientifiques, et découvrir le monde industriel comme des cultures étrangères. De par mon appétence pour la recherche et l'informatique, j'ai eu la chance de pouvoir venir effectuer mon stage au sein de l'entreprise **PHIMECA**, située à Clermont-Ferrand, et depuis longtemps en partenariat avec SIGMA.

Ce stage aura été l'occasion pour moi ... (à suivre)

Je souhaite remercier ... (à suivre)

Note: Ce document est autant un rapport de stage qu'une note à qui voudrait se servir des différentes méthodes explorées et codes développés.

1 Introduction

Dans le cadre d'analyse probabiliste et fiabiliste, il est d'usage d'effectuer des analyses de sensibilité sur une ou plusieurs grandeurs d'intérêt, grâce à divers moyens comme les indices de Sobol' ou les analyses de corrélation.

Ces analyses sont néanmoins souvent cantonnées à des modèles n'ayant en sortie et entrée que des grandeurs scalaires, pour lesquelles de nombreuses méthodes existent dans la littérature.

L'objectif du travail présenté ici, est d'effectuer ce type d'analyse sur des modèles ayant en entrée plusieurs champs aléatoires et des variables aléatoires scalaires et en sortie aussi des champs et grandeurs scalaires.

Pour compléter cela, il était bien sûr aussi demandé de d'abord créer un modèle simple gouverné par des champs stochastiques et de faire de l'analyse de sensibilité sur ce modèle.

Il sera d'abord présenté différentes méthodes présentes dans la littérature, avec leurs potentiels avantages et défauts.

Cette approche des l'analyse de sensibilité nous renvoie aussi sur des champs de recherche bien différents, car les méthodes peuvent autant venir du champ de la mécanique que de la géologie ou l'étude environnementale.

2 Recherche bibliographique

Ce travail de recherche a été l'occasion de parcourir un grand nombre de travaux, liés de plus ou moins loin à l'analyse de sensibilité sur des processus gaussiens. On notera notamment les travaux de [1] qui donnent dans leur travail un état de l'art complet sur les différentes méthodes utilisables pour l'analyse de sensibilité sur champs gaussiens.

D'autres techniques plus récentes ont néanmoins vu le jour, et ce sont ces dernières que nous avons explorés en premier. Notamment les travaux de [5] sur l'analyse de sensibilité de structures ayant en entrée des variables aléatoires et processus stochastiques variant dans le temps, ou encore le travail de [3] qui développe un métamodel entre l'entrée et sortie d'un modèle gouverné par des champs stochastiques.

2.1 Travaux retenus

Etant donné que les missions d'analyse de sensibilité et d'incertitudes se font classiquement en se basant sur la bibliothèque *openTURNS* développé par PHIMECA, le choix a été fait d'adapter techniques de recherche et méthodes déjà existantes dans l'API au problème étudié.

2.2 Methodes et outils de travail

L'ensemble des codes ont été écrits dans le langage *Python*, en se servant majoritairement des bibliothèques à caractère scientifique (*SciPy*, *NumPy*) et la bibliothèque développée en partie par PHIMECA : *openTURNS* (*open treatment of Uncertainties and Risk & Statistics*) [2].

PHIMECA a fourni un poste de travail Linux, tout comme un accès au serveur de calcul pour les modèles un peu plus conséquents.

2.3 Analyse de la problématique, premier tests et prise en main des outils

2.3.1 Mission confiée et tâches associées

L'objectif du stage était de développer une méthodologie pour l'analyse de sensibilité sur des modèles gouvernés par des champs aléatoires (Champs Stochastiques) et des variables aléatoires, et qui en sortie sont aussi représentables par une collection de champs et variables aléatoires.

Plus spécifiquement, cette méthodologie était à appliquer à un modèle d'échangeur de chaleur air-air commercialisé par **LIEBHERR** et utilisé pour la régulation thermique de l'air ambiant dans les avions de ligne.

Comme la mission confiée au sein de **PHIMECA** se rapprochait d'un travail de recherche, celui-ci était bien sur accompagné de nombreuses étapes distinctes. En effet, en plus du développement des codes et de l'application analytique, une recherche bibliographique a du être menée en amont pour parvenir à lister les différentes méthodes d'analyse existantes et tester celles qui pourraient convenir au mieux. De plus, une phase d'apprentissage et de tests a du d'abord être nécessaire pour parvenir à prendre en main les différents outils, et comprendre la théorie et les mathématiques sous-jacentes.

2.3.2 Prise en main d'openTURNS et de la théorie sur les champs stochastiques

Champs Stochastiques

Un champ stochastique est un champ de variables aléatoires toutes corrélées entre elles, et dont l'intensité de la corrélation est déterminée par leur proximité dans l'espace. Par exemple la position de tout objet est corrélée à sa position aux intervalles de temps proches (continuité du temps), ou bien les températures au dessus d'un pays sont corrélés à faible distance mais l'effet aléatoire est plus présent lorsque les distances sont importantes.

Ce type de modélisation des champs est couramment utilisée pour modéliser des

propriétés présentant des variabilités gaussiennes mais une continuité dans leur espace de définition.

Lorsque la corrélation est seulement dépendante de la distance entre deux points de l'espace et non d'une position absolue, on parle d'un champ stationnaire. Lors de cette étude l'on se place dans ce cas.



Figure 1: Réalisation d'un champ stochastique en deux dimension. Le champ est continu dans l'espace et la primitive est une gaussienne en deux dimensions.

Mathématiquement, cette corrélation peut être définie par différents modèles. Les deux modèles de corrélation les plus simples sont les fonctions de covariance exponentielles (1) et les exponentielles quadratiques (2).

$$C(d) = \exp(-d/V) \quad (1)$$

$$C(d) = \exp(-(d/V)^2); \quad (2)$$

Avec V étant le paramètre d'échelle et d la norme euclidienne de la distance entre deux points de l'espace considéré. *(plus le paramètre d'échelle est grand, plus le champ est lissé, tout comme le modèle quadratique est aussi plus lissé)* Une autre fonction de covariance, celle de *Matérn*, est aussi très utilisée, puisqu'elle présente des comportements limites similaires aux deux fonctions exponentielles présentes plus haut.

$$C_v(d) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{d}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{d}{\rho} \right); \quad (3)$$

La continuité des processus gaussiens et leur comportement quasi ondulatoire, permet de les traiter de manière un peu analogue à la décomposition de fourrier, et des les décomposer en une somme infinie de variables aléatoires gaussiennes décorréliées.. Cette méthode de décomposition vient du théorème de *Karhunen-Loève*.

La construction de cette série infinie se fait grâce aux vecteurs propres issus de la matrice de corrélation du modèle, et grâce à une base de vecteurs orthonormaux dans un espace Hilbertien. Les détails mathématiques de cette méthode peuvent être trouvées en [4].

Lors de l'approximation de *Karhunen-Loève*, la série est tronquée à l'ordre \mathbf{M} .

$H(\mathbf{x}, \theta)$	Approx. Procéssus Gaussien
λ_i	Valeur Propre de la matrice de covariance
ξ_i	Variable Normale Centrée Réduite
$\varphi_i(\mathbf{x})$	Vecteur propre de la matrice de covariance

$$H(\mathbf{x}, \theta) = \sum_{i=1}^M \sqrt{\lambda_i} \xi_i(\theta) \varphi_i(\mathbf{x}); \quad (4)$$

Cette décomposition permet de représenter l'ensemble de variabilité du champ avec des variables gaussiennes décorréées, et donc de faire des analyses de sensibilités sur cette nouvelle représentation du champ.

openTURNS

openTURNS est initialement un projet de bibliothèque open-source pour le traitement des incertitudes et des risques, commun a trois entreprises fondatrices, **Airbus**, **EDF** et **Phimeca Engineering**, projet ayant débuté en 2005. Depuis, deux autres organismes, **IMACS** et l'**ONERA** ont rejoint le développement d'*openTURNS*, qui se révèle être un grand atout pour le traitement des incertitudes et l'ingénierie fiabiliste. En effet, le projet d'*openTURNS* est un regroupement d'un ensemble d'algorithmes performants écrits en C++, se basant sur la théorie développée pour les traitement des incertitudes, l'optimisation robuste, et les études de sensibilité.

Un *wrapper* a été utilisée pour lier l'ensemble de la bibliothèque C++ à un module Python, et de pouvoir profiter de la facilité d'utilisation du langage Python et de sa flexibilité, tout en gardant les vitesses d'exécution du C++.

PHIMECA développe des logiciels commerciaux en se basant sur le module *openTURNS*, qui sont plus ergonomiques dans leur utilisation et automatisent certaines parties moins évidentes en utilisation directe de la librairie.

Néanmoins, la librairie reste quand même extrêmement bien documentée, avec la théorie mathématique sous-jacente à chaque algorithme d'explicitée, tout comme de nombreux exemples. La documentation présente de même des cas d'applications précis montrant la logique à avoir lors de l'écriture de codes avec *openTURNS*.

2.4 Exemple théorique simple, premiers codes

Pour parvenir a tester les méthodes trouvées lors de la recherche bibliographique, et avoir un modèle simple et rapide à exécuter, un exemple simple a été développé : Il s'agit d'un modèle de poutre en appui sur ses deux extrémités

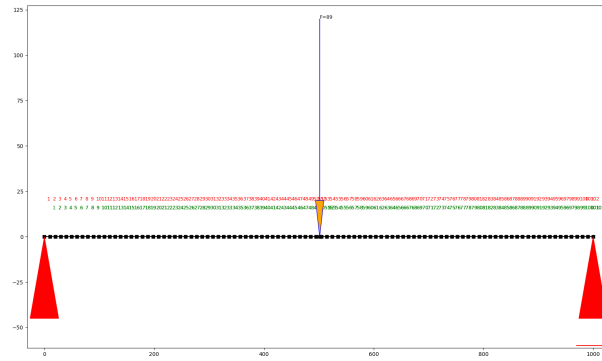


Figure 2: Modélisation d'une poutre en appui sur ses deux extrémités avec la bibliothèque anastruct

La particularité de ce modèle de poutre est, en plus d'être une aberration physique, le fait que le diamètre et le module de young sont gouvernés par un processus gaussien en une dimension, suivant la longueur de la barre. Pour parvenir à modéliser cette variation, le modèle est bien-sûr subdivisé en une centaine d'éléments finis, et le champ discrétisé sur un maillage de même longueur. En plus de ces deux grandeurs gouvernées par des champs, la densité du matériau, la position de la force, et la norme de la force dont représentés par des variables aléatoires Gaussiennes.

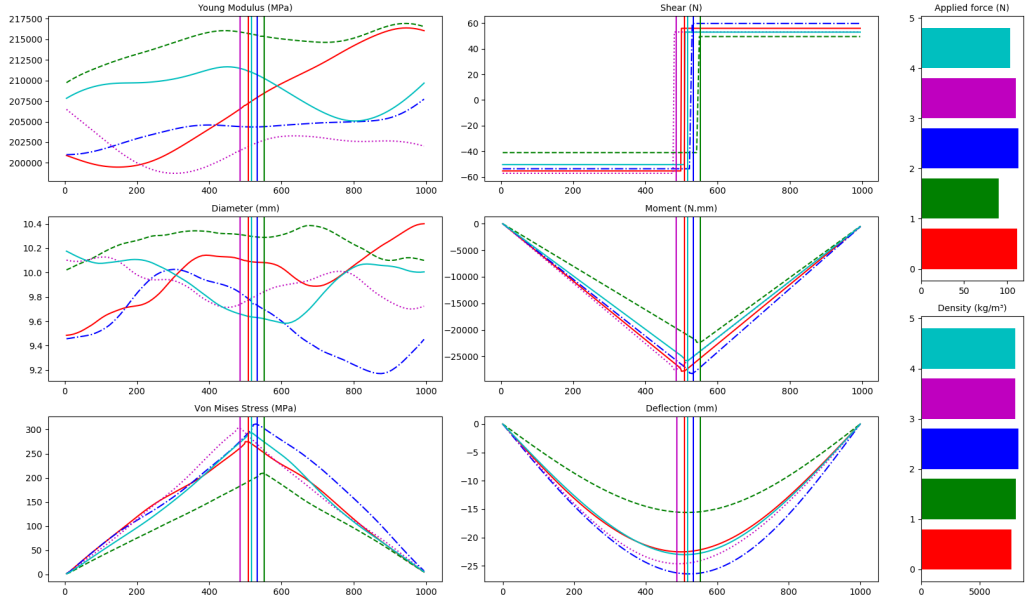


Figure 3: Échantillon de 5 réalisations de la poutre en flexion. [Sur la gauche : Processus gouvernant le module de young, Processus gouvernant le diamètre, contrainte de Von Mises] | [Sur la droite : Contrainte de cisaillement, moment de flexion, déflexion]

Les grandeurs d'étude choisies ici sont la contrainte de Von Mises, la déflexion de chaque noeud, et la déflexion maximale. On pourrait de même choisir une contrainte de Von Mises maximale pour représenter la défaillance.

Comme chacun des processus stochastiques peut être décomposé en une somme finie de variables aléatoires normales grâce à la décomposition de karhunen-Loeve, l'on peut à l'inverse, générer des champs stochastiques à partir d'une collection de variables aléatoires, dont les paramètres sont issus de l'analyse même du champ.

L'on considère donc notre modèle plus comme fonction de variables aléatoires et de processus stochastiques, mais seulement comme une fonction de variables aléatoires scalaires.

Ceci permet de faire l'analyse de sobol classique (avec l'algorithme de sensibilité de *Saltelli*) sur ces nouvelles variables aléatoires.

La difficulté ici, est d'ensuite relier les indices de sensibilité de Sobol des variables aléatoires définissant le champ au champ stochastique même. (On ne souhaite n'avoir qu'un seul indice de sobol pour un champ.)

2.5 Choix des codes et difficultés algorithmiques

Au vu de choix de développer une méthode algorithmique pour l'analyse de sensibilité sur des champs stochastiques, et non pas de juste faire l'analyse de sensibilité sur une seule problématique, il y avait un besoin de robustesse supplémentaire pour l'écriture des différents codes. Ces derniers doivent pouvoir fonctionner avec tout type de fonction en python prenant en entrée des champs et variables aléatoire, et renvoyant un tuple de variables aléatoires.

Pour pouvoir tester ceci, l'exemple de la poutre en éléments fins à été codée a part, et peut être utilisée directement comme fonction, indépendamment de l'analyse de sensibilité. Cette dernière prend de même en entrée deux champs stochastiques et trois variables aléatoires, et peut renvoyer plusieurs arguments. Les tests ont été faits avec la fonction renvoyant qu'une collection de champs (contraintes de von mises) et avec la fonction renvoyant un tuple (collection de champs + vecteur déflexion maximale). Ce choix augmente le nombre de vérification qu'il y a effectuer au sein du code, mais permet l'utilisation d'un plus grand nombre de fonctions, et l'analyse de sensibilité sur l'ensemble des variables de sortie en une seule fois.

Pour parvenir à bien contrôler les processus stochastiques, une classe a été écrite *NdGaussianProcessConstructor*, qui permet de définir entièrement un processus stochastique, et possède de nombreuses méthodes pour créer des samples, faire les décompositions de Karhunen-Loeve, ou encore reconstituer un champ à partir de la dite décomposition. Enfin, comme les samples sont nécessaires pour la décomposition de Karhunen-Loeve, et que ceux-ci peuvent être de taille plutôt importante, ils sont enregistrés sous forme de *numpy.memmap* dans un fichier temporaire, pour relacher un peu de pression sur la mémoire vive.

En interne, le modèle fonction de champs stochastiques, passe par un wrapper qui fait qu'on peut la considérer comme seulement fonction de variables aléatoires gaussiennes. Cela permet d'utiliser les méthodes internes à openturns comme l'algorithme de saltellie pour identifier les indices de sobol.

Néanmoins, pour pallier à certaines difficultés comme les fonctions qui par moment renvoient des *np.nan* (donc lorsqu'il y a eu une erreur), on passe d'abord par une étape de correction: l'ensemble des valeurs de sortie contenant des *nan* sont régénérés, et comme les entrées de l'étude sont des combinaisons de deux ensembles de variables aléatoires A et B, toutes les autres permutations des entrées ayant entraîné l'erreur sont supprimées et remplacées.

Les codes sont accessibles via github: https://github.com/Kramer84/stochastic_process_analysis

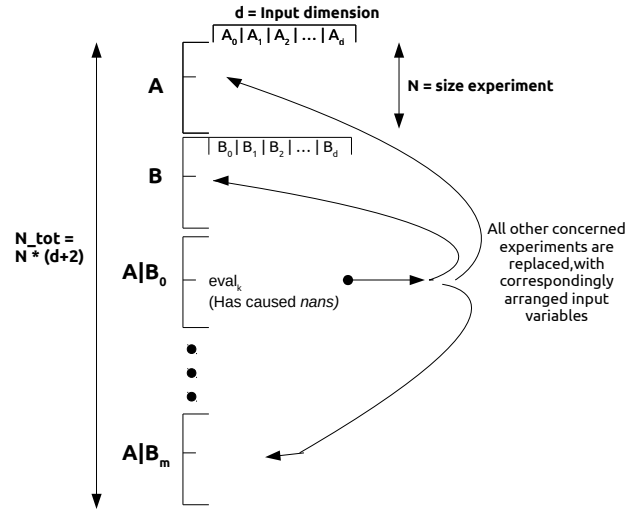


Figure 4: Organsiation des variables d'entrée, expliquant l'étape de postprocessing

\LaTeX uses a simple and convenient system for assigning numbered labels to equations and other objects (figures, tables, etc. . .) and for referring to them. After having edited the source file and rearranged the position of the equations, \LaTeX will change labels and references consistently throughout the text (if you did the things right of course. . .)

Examples of text containing mathematical expressions and equations:

M_r	mass internal to the radius r
m	mass of the zone
r_0	unperturbed zone radius
ρ_0	unperturbed density in the zone
T_0	unperturbed temperature in the zone
L_{r0}	unperturbed luminosity
E_{th}	thermal energy of the zone

$$\tau_{\text{co}} = \frac{E_{\text{th}}}{L_{r0}}, \quad (5)$$

$$\tau_{\text{ff}} = \sqrt{\frac{3\pi}{32G} \frac{4\pi r_0^3}{3M_r}}, \quad (6)$$

$$\nabla_{\text{ad}} = \left(\frac{\partial \ln T}{\partial \ln P} \right)_S, \quad \chi_T = \left(\frac{\partial \ln P}{\partial \ln T} \right)_\rho, \quad \kappa_T = \left(\frac{\partial \ln \kappa}{\partial \ln T} \right)_T$$

$$\frac{\pi^2}{8} \frac{1}{\tau_{\text{ff}}^2} (3\Gamma_1 - 4) > 0 \quad (7)$$

$$\frac{\pi^2}{\tau_{\text{co}} \tau_{\text{ff}}^2} \Gamma_1 \nabla_{\text{ad}} \left[\frac{1 - 3/4 \chi_\rho (\kappa_T - 4) + \kappa_P + 1}{\chi_T} \right] > 0 \quad (8)$$

$$\frac{\pi^2}{4} \frac{3}{\tau_{\text{co}} \tau_{\text{ff}}^2} \Gamma_1^2 \nabla_{\text{ad}} \left[4\nabla_{\text{ad}} - (\nabla_{\text{ad}} \kappa_T + \kappa_P) - \frac{4}{3\Gamma_1} \right] > 0 \quad (9)$$

2.6 Example of verbatim text

```
PROGRAM area
REAL base, height, area
PRINT *, 'Enter the values for the base and height of a triangle.'
READ *, base, height
area = (1.0/2.0) * base * height
PRINT *, 'The area of a triangle with base ', base
PRINT *, 'and height ', height, ' is ', area
STOP
END
```

Note: In *verbatim* mode you can easily end up outside the margins, as in the example above: pay attention to that!

2.7 Lists

Example of a list with numbered items:

1. Planets, asteroids, moons ...
2. Stars, galaxies, quasars

Example of a list with unnumbered items:

- Planets, asteroids, moons ...
- Stars, galaxies, quasars

3 Results

4 Tables and figures

Table 1: Example of table caption: opacity sources.

Source	$T/[\text{K}]$
Yorke 1979, Yorke 1980a	≤ 1700
Krügel 1971	$1700 \leq T \leq 5000$
Cox & Stewart 1969	$5000 \leq$

5 Discussion

6 Conclusions

References

- [1] Linda Lilburne and Stefano Tarantola. Sensitivity analysis of spatial models. *International Journal of Geographical Information Science*, 23(2):151–168, Feb 2009.
- [2] Bertrand Iooss Michaël Baudin, Anne Dutfoy and Anne-Laure Popelin. Openturns: An industrial software for uncertainty quantification in simulation. 2015.
- [3] Luc Pronzato. Sensitivity analysis via Karhunen–Loève expansion of a random field model: Estimation of Sobol’ indices and experimental design. *Reliab. Eng. Syst. Saf.*, 187:93–109, Jul 2019.
- [4] Bruno Sudret and Armen Der Kiureghian. Stochastic Finite Element Methods and Reliability A State-of-the-Art Report. *ResearchGate*, Jan 2000.
- [5] Pengfei Wei, Yanyan Wang, and Chenghu Tang. Time-variant global reliability sensitivity analysis of structures with both input random variables and stochastic processes. *Struct. Multidiscip. Optim.*, 55(5):1883–1898, May 2017.