

Couplage entre indices à base de dérivées et mode adjoint pour l'analyse de sensibilité globale. Application sur le code Mascaret

Sébastien Petit, Fabrice Zaoui, Anne-Laure Popelin, Cédric Goeury, Nicole Goutal

▶ **To cite this version:**

Sébastien Petit, Fabrice Zaoui, Anne-Laure Popelin, Cédric Goeury, Nicole Goutal. Couplage entre indices à base de dérivées et mode adjoint pour l'analyse de sensibilité globale. Application sur le code Mascaret. 2016. hal-01373535

HAL Id: hal-01373535
https://hal.science/hal-01373535

Preprint submitted on 28 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Soumission

Journal de la Société Française de Statistique

arXiv: math.PR/00000000

Couplage entre indices à base de dérivées et mode adjoint pour l'analyse de sensibilité globale. Application sur le code Mascaret.

Sébastien Petit[1], Fabrice Zaoui[1], Anne-Laure Popelin[2], Cédric Goeury[1] et Nicole Goutal[1]

Résumé : Dans ce papier, nous proposons une méthodologie d'analyse de sensibilité globale pour pallier les problèmes de coût d'estimation des indices de Sobol' dans les problèmes de grande dimension. Cette méthodologie est basée sur l'utilisation jointe de la différentiation automatique du code de calcul et de mesures de sensibilité basées sur les dérivées (dénommées DGSM) permettant de borner les indices totaux de Sobol'. Nous montrons l'efficacité de cette approche sur un cas d'étude industriel, utilisant le code de calcul hydraulique Mascaret. Nous discutons de plus de l'efficacité des DGSM pour la détection d'interactions.

Abstract: In this paper we propose a global sensitivity analysis procedure which aims to remedy the cost estimation problems of Sobol' indices for high-dimensional models. This procedure is based on the use of derivative based global sensivitiy measures (DGSM) and automatic differentiation. This allows us to obtain bounds for total Sobol' indices. We show the efficacity of this approach on an industrial case : the hydrodynamic simulation code Mascaret. We also discuss the use of derivatives based indices for the detection of interactions.

Mots-clés : Analyse de sensibilité, Indices de Sobol', DGSM, Différentiation automatique, Mascaret
Keywords: Sensitivity Analysis, Sobol' Indices, DGSM, Automatic Differentiation, Mascaret
Classification AMS 2000 : 35L05, 35L70

1. Introduction

Analyser un modèle en termes de sensibilité consiste à étudier l'influence des incertitudes sur ses entrées sur la variabilité de sa sortie. On distingue l'analyse de sensibilité locale, effectuée souvent à l'aide de dérivées autour d'un point de fonctionnement, de l'analyse de sensibilité globale où l'on évalue l'effet d'un facteur en faisant également varier les autres, ce qui permet de rendre notamment compte d'effets d'interaction. Pour un état de l'art des méthodes d'analyse de sensibilité globale, voir [Iooss and Lemaître, 2015].

Le calcul des indices de Sobol' est une technique très répandue en analyse de sensibilité globale. Cette méthode introduite par Sobol' ([Sobol, 1990]) puis améliorée par Saltelli ([Saltelli, 2002]) fait partie des méthodes dites d'analyse de la variance. Comme ce nom l'indique, cette méthode

[1] EDF R&D : Laboratoire National d'Hydraulique et Environnement
E-mail : sebastien.julien.petit@gmail.com
E-mail : cedric.goeury@edf.fr
E-mail : nicole.goutal@edf.fr
E-mail : fabrice.zaoui@edf.fr
[2] EDF R&D : Département de Management des Risques Industriels
E-mail : anne-laure.popelin@edf.fr

Soumis au Journal de la Société Française de Statistique
File: jsfds-manuel_utilisateurs.tex, compiled with jsfds, version : 2009/12/09
date: May 21, 2016

consiste à décomposer la variance de la sortie du modèle en parts de variance, appelées indices de Sobol', expliquées par les différents sous-groupes de variables d'entrée. Ces indices permettent donc de rendre compte, d'une part de l'influence d'une variable au premier ordre et d'autre part d'effets d'interaction à tout ordre.

L'approche classique consiste à calculer les indices de Sobol de manière itérative : en premier lieu sont calculés les indices du premier ordre et des indices dits totaux pour chacune des variables, puis éventuellement des indices d'ordres supérieurs afin de quantifier finement l'impact des interactions. Les indices totaux quantifient pour chaque entrée la part de variance expliquée en prenant en compte son effet individuel et toutes les interactions dans lesquelles elle est impliquée. Les indices totaux et du premier ordre sont le plus souvent estimés conjointement par une méthode de type Monte-Carlo ([Saltelli, 2002]). L'inconvénient de cette procédure est son coût en nombre de simulations i.e. d'appels au modèle. En effet $(d+2)n$ simulations sont requises où d désigne la dimension du problème et n une taille d'échantillons ($n = 10^4$ permet d'obtenir une bonne estimation pour la plupart des applications). Dans le cas de modèles coûteux en temps de calcul unitaire, et présentant un nombre important de variables d'entrée, ce coût peut se révéler rédhibitoire.

Dans cet article, nous discutons de techniques permettant de réaliser en partie cette première étape pour un coût (i.e. un nombre de simulations) non proportionnel à d.

Plus précisément nous cherchons à borner les indices totaux de Sobol' en calculant des indices à base de dérivées dits DGSM (Derivated based Global Sensitivity Measures, voir [Sobol and Kucherenko, 2009], [Lamboni et al., 2013], [Kucherenko and Song, 2015] et [Kucherenko and Iooss, 2015]). Pour atteindre nos objectifs de coût nous calculons ces indices à base de dérivées en utilisant un mode adjoint du modèle. L'intérêt du mode adjoint, qui est une implémentation particulière de la règle de la dérivée composée (voir [Griewank and Walther, 2008]) est ici de permettre le calcul d'un gradient en une seule exécution, 4 à 6 fois plus longue qu'une exécution du modèle initial. Le calcul d'indices à base de dérivées est par conséquent réduit d'un facteur théoriquement compris entre $\frac{d+1}{6}$ et $\frac{d+1}{4}$ par rapport à une estimation par différences finies. En particulier cette procédure nous permet de cribler en gagnant un temps considérable par rapport à une méthode de Morris.

En complément nous estimons les indices du premier ordre par lissage ([Da Veiga et al., 2009]), ce qui nécessite un nombre d'appels au modèle non proportionnel à d (10^4 appels permettent d'obtenir une bonne estimation pour la plupart des applications).

Nous présentons aussi un cas d'application de ces travaux : le logiciel opensource Mascaret, développé par EDF (pour plus d'informations, consulter http://www.opentelemac.org/), permettant de calculer des hauteurs d'eau lors d'écoulements fluviaux en fonction de paramètres physiques.

2. Présentation et discussion méthodologique

2.1. Notations

Soit $\mathbf{X} = (X_1, ..., X_d)$ un vecteur d'entrée aléatoire, de composantes indépendantes, de loi absolument continue par rapport à la mesure de Lebesgue, dont on note la densité $f = \prod_{i=1}^{d} f_i$, $F = \prod_{i=1}^{d} F_i$ la fonction de répartition et $\chi = \prod_{i=1}^{d} I_i$ le support, en supposant que les I_i sont des intervalles. Nous considérons dans cet article un modèle déterministe $G \in C^1(\chi, \mathbb{R}) \cap L^2(dF)$ tel

que $\frac{\partial G}{\partial x_i} \in L^2(dF)$ pour tout i. Nous notons $Y = G(\mathbf{X})$ la variable d'intérêt en sortie.

On note $P_d^1 = P_d \setminus \{\emptyset\}$ où P_d est l'ensemble des parties de $[\![1,d]\!]$. Pour $\mathbf{x} = (x_1, ..., x_d)$, un vecteur et $J \in P_d^1$, on désigne par $\mathbf{x}_J$ le vecteur dont les composantes sont les x_j, pour $j \in J$, rangés en ordre croissant par rapport à j. De plus, pour $J_1, J_2 \in P_d^1$, nous noterons $(\mathbf{x}_{J_1}, \mathbf{x}_{J_2}) = \mathbf{x}_{J_1 \cup J_2}$.

Enfin dans tout cet article nous désignerons par i la i-ème variable d'entrée du modèle et nous écrirons $-i = [\![1,d]\!] - i$.

Cette section méthodologique s'organise en trois sous-sections comme suit :

- Décomposition Anova et indices de Sobol',
- Différentiation automatique,
- Indices à base de dérivées DGSM.

2.2. Décomposition Anova et indices de Sobol'

2.2.1. Définition

En supposant $G \in L^2(dF)$, nous sommes assurés de l'existence et de l'unicité de la décomposition suivante ([Efron and Stein, 1981]) :

$$G(\mathbf{x}) = g_0 + \sum_{J \in P_d^1} g_J(\mathbf{x}_J), \tag{1}$$

où

$$g_0 = \int G dF \text{ et } g_J \in L^2(dF_J), \forall J \in P_d^1 \tag{2}$$

sous les contraintes

$$\int g_J(\mathbf{x}_J) dF_i(x_i) = 0 \; \forall J \in P_d^1 \; \forall i \in J \; \forall \mathbf{x}_{J-\{i\}}. \tag{3}$$

Nous avons par conséquent

$$\int g_J g_K dF = 0 \; \forall J, K \in P_d^1, J \neq K, \tag{4}$$

et

$$\text{var}(Y) = \sum_{J \in P_d^1} \text{var}(g_J(\mathbf{X}_J)). \tag{5}$$

Nous notons alors les indices de Sobol'

$$S_J = \frac{\text{var}(g_J(\mathbf{X}_J))}{\text{var}(G(\mathbf{X}))}, \text{ pour } J \in P_d^1. \tag{6}$$

On remarque que pour d variables, il existe $2^d - 1$ indices à calculer, ce qui devient rapidement irréalisable lorsque d augmente.

2.2.2. *Indices d'intérêt*

Comme évoqué en introduction, nous nous intéressons au calcul, pour chaque entrée i, de l'indice du premier ordre

$$S_i = S_{\{i\}},$$

et de l'indice dit total

$$S_{T_i} = \sum_{J \subset [\![1,d]\!], i \in J} S_J = \frac{\int u_i^2 dF}{\mathrm{var}(Y)},$$

où $u_i = \sum_{J \subset [\![1,d]\!], i \in J} g_J$.

L'indice du premier ordre S_i représente la part de la variance de Y apportée par X_i, sans prendre en compte les éventuels effets d'interaction. Ces indices constituent donc une analyse au premier ordre mais ne suffisent en général pas à expliquer entièrement la variance (voir (5)) :

$$\sum_{i=1}^{d} S_i \leq 1.$$

L'indice total S_{T_i} représente la part de variance expliquée par X_i, en prenant en compte tous les effets d'interaction dans lesquels elle est impliquée (la part d'interaction vaut donc $S_{T_i} - S_i$). En sommant les indices totaux, on prend en compte plusieurs fois les mêmes interactions et on a donc (voir (5)) :

$$\sum_{i=1}^{d} S_{T_i} \geq 1.$$

Les indices totaux nous permettent d'orienter nos éventuels calculs d'indices d'ordres supérieurs. En effet nous n'allons calculer que les indices associés à des variables impliquées dans des interactions (c'est-à-dire lorsque $S_{T_i} - S_i$ est "important").

2.3. Différentiation automatique

Dans cette section nous cherchons à calculer des gradients de manière précise et rapide. Pour un état de l'art plus complet voir [Griewank and Walther, 2008]).

On peut voir un code numérique qui prend d variables en entrée pour en calculer q autres en sortie comme une fonction $\mathbf{f} : \mathbb{R}^d \to \mathbb{R}^q$. Si le code numérique possède N lignes et si l'on note $(\mathbf{f}_k)_{1 \leq k \leq N}$ les fonctions mathématiques correspondant aux transformations sur les variables, on a la relation :

$$\mathbf{f} = \mathbf{f}_N \circ \ldots \circ \mathbf{f}_1. \tag{7}$$

Ainsi si $\mathbf{f}$ et les $\mathbf{f}_k$ sont différentiables on a, en notant J une jacobienne :

$$J_{\mathbf{f}}(\mathbf{x}) = J_{\mathbf{f}_N}(\mathbf{f}_{N-1} \circ \ldots \circ \mathbf{f}_1(\mathbf{x})) * \ldots * J_{\mathbf{f}_1}(\mathbf{x}).$$

Les transformations appliquées sur les lignes étant simples, on peut souvent calculer leur jacobiennes analytiquement. Si l'on parvient à le faire pour chaque ligne, il suffit ensuite de les multiplier pour obtenir la jacobienne analytique du code.

En pratique on calcule des deltas successifs plutôt que les jacobiennes complètes, comme c'est implémenté dans le logiciel Tapenade développé par l'Inria
(voir http://www-sop.inria.fr/tropics/tapenade.html). Tapenade possède un dérivateur formel capable de différentier la plupart des opérations qui peuvent être implémentées en une ligne de code. L'implémentation, très efficace, propose les modes tangent et adjoint.

2.3.1. Mode tangent

Le mode tangent consiste à calculer le delta en sortie correspondant à un delta en entrée

$$\Delta \mathbf{y} = J_{\mathbf{f}}(\mathbf{x})\Delta \mathbf{x} = J_{\mathbf{f}_N}(\mathbf{f}_{N-1} \circ ... \circ \mathbf{f}_1(\mathbf{x})) * ... * J_{\mathbf{f}_1}(\mathbf{x})\Delta \mathbf{x},$$

en multipliant la prochaine jacobienne au delta courant de proche en proche lorsqu'on parcours le code dans le sens de l'éxécution. On calcule donc à la ligne $k+1$:

$$\Delta(\mathbf{f}_{k+1} \circ ... \circ \mathbf{f}_1(\mathbf{x})) = J_{\mathbf{f}_{k+1}}(\mathbf{f}_k \circ ... \circ \mathbf{f}_1(\mathbf{x}))\Delta(\mathbf{f}_k \circ ... \circ \mathbf{f}_1(\mathbf{x})).$$

Ainsi, pour $\Delta x_j = \delta_{ij}$, nous avons $\Delta \mathbf{y} = {}^t(\frac{\partial f_1(\mathbf{x})}{\partial x_i}, ..., \frac{\partial f_q(\mathbf{x})}{\partial x_i})$. Nous obtenons donc une colonne de la jacobienne et donc les dérivées partielles de toutes les sorties par rapport à une entrée.

2.3.2. Mode Adjoint

L'approche consiste à permuter les rôles des entrées et des sorties. Pour un delta $\Delta \mathbf{y}$ sur la sortie initiale $\mathbf{y}$, on construit :

$$\Delta \mathbf{x} = {}^t J_{\mathbf{f}}(\mathbf{x})\Delta \mathbf{y} = {}^t J_{\mathbf{f}_1}(\mathbf{x}) * ... * {}^t J_{\mathbf{f}_N}(\mathbf{f}_{N-1} \circ ... \circ \mathbf{f}_1(\mathbf{x}))\Delta \mathbf{y},$$

en multipliant la transposée de la jacobienne de la ligne précédente au delta courant de proche en proche lorsque l'on parcours le code dans le sens inverse de l'exécution. On calcule donc à la ligne k :

$$\Delta(\mathbf{f}_k \circ ... \circ \mathbf{f}_1(\mathbf{x})) = {}^t J_{\mathbf{f}_{k+1}}(\mathbf{f}_k \circ ... \circ \mathbf{f}_1(\mathbf{x}))\Delta(\mathbf{f}_{k+1} \circ ... \circ \mathbf{f}_1(\mathbf{x})).$$

Ainsi, pour $\Delta y_j = \delta_{ij}$, on a $\Delta \mathbf{x} = {}^t(\frac{\partial f_i(\mathbf{x})}{\partial x_1}, ..., \frac{\partial f_i(\mathbf{x})}{\partial x_d})$. Nous obtenons alors une colonne de la transposée de la jacobienne, donc une ligne de la jacobienne, c'est-à-dire le gradient d'une sortie par rapport aux entrées.

Notons que pour calculer les jacobiennes il est nécessaire de connaître les valeurs des paramètres successifs $\mathbf{f}_k \circ ... \circ \mathbf{f}_1(\mathbf{x})$ et donc d'exécuter le code initial. Ces problèmes sont gérés par Tapenade, notamment via une structure de piles pour le mode adjoint.

Dans notre cas d'application la sortie est scalaire et nous avons donc choisi le mode adjoint, qui nous permet de calculer des gradients de manière très efficace. Le code adjoint donne un gradient en une seule exécution 4 à 6 fois plus longue théoriquement que le code initial ([Griewank and Walther, 2008]), et ce quelle que soit la dimension. Cette méthode est donc largement plus efficace qu'un calcul de dérivées par différences finies.

Auteur	hypothèse	Inf ou Sup	
[Lamboni et al., 2013]	continuité de f_i	Sup	$\mathrm{UB}_1(i) = \frac{C(f_i)}{\mathrm{var}(Y)} \int \left(\frac{\partial G}{\partial x_i}\right)^2 dF$
[Sobol and Kucherenko, 2009]	$\left\lvert\frac{\partial G}{\partial x_i}\right\rvert \leq C$	Sup	$\mathrm{SK}_{\mathrm{Sup}}(i) = \frac{\sigma_i^2 C^2}{\mathrm{var}(Y)}$
[Sobol and Kucherenko, 2009]	$c \leq \left\lvert\frac{\partial G}{\partial x_i}\right\rvert$	Inf	$\mathrm{SK}_{\mathrm{Inf}}(i) = \frac{\sigma_i^2 c^2}{\mathrm{var}(Y)}$
[Kucherenko and Song, 2015]	X_i uniforme sur $[0,1]$	Sup	$\mathrm{UB}_2(i) = \frac{\int x_i(1-x_i)\left(\frac{\partial G}{\partial x_i}(\mathbf{x})\right)^2 dF(\mathbf{x})}{2\mathrm{var}(Y)}$
[Kucherenko and Song, 2015]	X_i uniforme sur $[0,1]$	Inf	$\mathrm{LB}_1(i) = \frac{\left(\int (G(1,\mathbf{x}_{-i}) - G(0,\mathbf{x}_{-i}))(G(1,\mathbf{x}_{-i}) + G(0,\mathbf{x}_{-i}) - 2G(\mathbf{x}))dF(\mathbf{x})\right)^2}{4\mathrm{var}(Y)\int\left(\frac{\partial G}{\partial x_i}\right)^2 dF}$
[Kucherenko and Iooss, 2015]	$X_i \sim N(m_i, \sigma_i^2)$	Inf	$\mathrm{LB}_2(i) = \frac{\sigma_i^4}{(m_i^2+\sigma_i^2)\mathrm{var}(Y)}\left(\int \frac{\partial G}{\partial x_i} dF\right)^2$
[Kucherenko and Song, 2015]	X_i uniforme sur $[0,1]$	Inf	$\mathrm{LB}_3(i) = \frac{(2m^*+1)\left[\int\left(G(1,\mathbf{x}_{-i}) - G(\mathbf{x}) + x_i^{m^*+1}\frac{\partial G}{\partial x_i}(\mathbf{x})\right)dF(\mathbf{x})\right]^2}{(m^*+1)^2\mathrm{var}(Y)}$

TABLE 1. ***État de l'art sur les DGSM. Le m^* figurant dans l'expression de la borne $\mathrm{LB}_3(i)$ correspond à un m optimal que nous recherchons via une méthode d'optimisation simple. Cela ne nécessite aucun appel au modèle supplémentaire. Les hypothèses nécessaires venant compléter celles faites dans le cadre de notre étude sont renseignées dans le champ 'hypothèse'.***

2.4. Derivated based Global Sensitivity Measures (DGSM)

Il existe également des méthodes basées sur les dérivées permettant de déduire des informations de sensibilité globales à partir d'indices de sensibilité locaux. Nous nous intéressons dans le cadre de ces travaux aux indices dits DGSM (Derivated based Global Sensitivity Measures) et cherchons à encadrer les indices totaux de Sobol' via ces derniers.
Une borne supérieure égale à l'indice du premier ordre nous permet de détecter une absence d'interactions ou plus simplement d'éliminer une variable non influente lorsque la borne est nulle. Une borne inférieure plus grande que S_i nous renseigne sur la présence d'interactions.

2.4.1. État de l'art

Nous avons supposé que G était de classe C^1, par conséquent u_i l'est aussi et :

$$\frac{\partial G}{\partial x_i} = \frac{\partial u_i}{\partial x_i}. \tag{8}$$

On trouve dans la littérature des bornes DGSM basées sur ce résultat (voir [Sobol and Kucherenko, 2009], [Lamboni et al., 2013], [Kucherenko and Song, 2015] et [Kucherenko and Iooss, 2015]). Nous notons Inf pour une borne inférieure et Sup pour une borne supérieure et présentons les résultats de la littérature dans le tableau 1.

On remarque que l'indice UB_1 fourni un résultat similaire à l'évaluation de la moyenne des effets élémentaires absolus dans le cadre d'une méthode de Morris. Notons de plus que les indices valables pour les lois uniformes sur $[0,1]$ se généralisent aisément à toutes les lois uniformes en normalisant.

Nous avons adapté le cas gaussien pour obtenir le cas gaussien tronqué.

Proposition 1. *Si X_i suit une distribution normale de moyenne m_i, de variance σ_i^2, tronquée sur $[m_i - a_i, m_i + a_i]$:*

$$S_{T_i} \geq \mathrm{LB}_4(i) = \frac{\sigma_i^4}{\mathbb{E}(X_i^2)\mathrm{var}(Y)} \left(\mathbb{E}(\frac{\partial G}{\partial x_i}(\mathbf{X})) - f_i(m_i + a_i)(\mathbb{E}(G(\mathbf{X})|X_i = m_i + a_i) - \mathbb{E}(G(\mathbf{X})|X_i = m_i - a_i)) \right)^2.$$

Des formules analytiques existent pour $\mathbb{E}(X_i^2)$.

2.4.2. Coût

Une étude de complexité montre que, comme évoqué précédemment, le calcul des dérivées par différences finies ne permet pas d'atteindre notre objectif en termes de temps de calcul. Par exemple, pour obtenir la première borne UB_1 présentée, il faut par différences finies effectuer $(d+1)n$ appels au modèle, où n est une taille d'échantillon.

La procédure de lissage ([Da Veiga et al., 2009]) implique l'estimation des fonctions $x \mapsto \mathbb{E}(G(\mathbf{X})|X_i = x)$. En utilisant ces méta-modèles, nous pouvons calculer les espérances conditionnelles aux bords des intervalles. Ces valeurs sont nécessaires pour le calcul des indices LB_1, LB_3 et LB_4. Le bon comportement de ces méta-modèles aux bords est prouvé dans [Fan and Gijbels, 1992]. Avec l'aide de ces méta-modèles et du mode adjoint nous pouvons estimer tous les indices présentés dans le tableau 1, excepté LB_1, en un nombre de simulations non linéairement dépendant de d.

2.4.3. Discussion

Nous avons cherché à évaluer les minorations données par les indices LB_2, LB_3 et LB_4. Malheureusement, ces estimateurs ne permettent pas d'évaluer les interactions.

Plus précisément :

Proposition 2. *Les trois bornes inférieures notées LB_2, LB_3 et LB_4 sont en réalité inférieures aux indices du premier ordre.*

Preuve. Ces bornes, que nous noterons LB, sont obtenues en appliquant l'inégalité de Cauchy-Schwartz sur l'espérance de $L(X_i)u_i(\mathbf{X})$ où $L \in L^2(dF_i)$.
Nous avons

$$S_i^{tot} = \frac{\int u_i^2 dF}{\mathrm{var}(Y)} \geq \frac{(\int L(x_i)u_i(\mathbf{x})dF(\mathbf{x}))^2}{\mathrm{var}(Y)(\int L^2 dF_i)} = \mathrm{LB}(i).$$

Par ailleurs

$$\int L(x_i)u_i(\mathbf{x})dF(\mathbf{x}) = \int L(x_i)g_i(x_i)dF_i(x_i) + \sum_{J \subset [\![1,d]\!], i \in J, |J| \geq 2} \int L(x_i) \left(\int g_J(\mathbf{x}_J)dF_{J-\{i\}}(\mathbf{x}_{J-\{i\}}) \right) dF_i(x_i).$$

$$= \int L(x_i)g_i(x_i)dF_i(x_i),$$

les termes d'interaction étant nuls à cause de (3). Nous avons donc

$$\mathrm{LB}(i) = \frac{(\int L(x_i)u_i(\mathbf{x})dF(\mathbf{x}))^2}{\mathrm{var}(Y)(\int L^2(x_i)dF_i(x_i))} = \frac{(\int L(x_i)g_i(x_i)dF_i(x_i))^2}{\mathrm{var}(Y)(\int L^2(x_i)dF_i(x_i))}$$

$$\leq \frac{\int g_i^2(x_i)dF_i(x_i)}{\mathrm{var}(Y)} = S_i,$$

grâce à l'inégalité de Cauchy-Schwarz appliquée à l'espérance de $L(X_i)g_i(X_i)$.

Les termes de u_i représentant les interactions :

$$\sum_{J\subset[\![1,d]\!],i\in J,|J|\geq 2} g_J,$$

sont orthogonaux à une fonction ne dépendant que de x_i et l'information sur les interactions est donc perdue dans le produit scalaire. Notons de plus que ces bornes inférieures ne sont pas moins coûteuses à calculer que les indices du premier ordre.

Nous proposons donc une piste pour étendre cette méthode, en calculant une borne inférieure de $S_{T_i} - S_i$. On s'affranchit ainsi de l'erreur commise sur S_i.

Proposition 3. *Soit $h : \prod_{j=1,j\neq i}^{d} I_j \to \mathbb{R}$, si f_i est de classe C^1 et si les hypothèses suivantes sont vérifiées :*

$$h \in L^2(dF_{-i}), \int h^2 dF_{-i} > 0, \tag{9}$$

$$\frac{f_i'}{f_i} \in L^2(dF_i), \int \left(\frac{f_i'}{f_i}\right)^2 dF_i > 0, \tag{10}$$

$$\int h dF_{-i} = 0, \tag{11}$$

$$\frac{\partial G}{\partial x_i} \in L^1(dF), \tag{12}$$

$$f_i(x_i)G(\mathbf{x}) \underset{x_i\to a_i,b_i}{\longrightarrow} 0, \forall \mathbf{x}_{-i}, \tag{13}$$

où a_i et b_i sont les bornes de I_i ($-\infty \leq a_i < b_i \leq +\infty$), alors

$$S_{T_i} - S_i \geq \mathrm{LBI}(h,i) = \frac{\left(\int \frac{\partial G}{\partial x_i}(\mathbf{x})h(\mathbf{x}_{-i})dF(\mathbf{x})\right)^2}{\mathrm{var}(Y)\int \left(\frac{h(\mathbf{x}_{-i})f_i'(x_i)}{f_i(x_i)}\right)^2 dF(\mathbf{x})}.$$

Preuve. D'après (9), (10), (11) nous avons

$$\int \frac{h(\mathbf{x}_{-i})u_i(\mathbf{x})f_i'(x_i)}{f_i(x_i)}dF(\mathbf{x}) = \int \frac{h(\mathbf{x}_{-i})(u_i(\mathbf{x}) - g_i(x_i))f_i'(x_i)}{f_i(x_i)}dF(\mathbf{x}).$$

En utilisant l'inégalité de Cauchy-Schwartz sur l'espérance de $\frac{h(\mathbf{X}_{-i})(u_i(\mathbf{X})-g_i(X_i))f_i'(X_i)}{f_i(X_i)}$ nous pouvons donc écrire :

$$\left(\int \frac{h(\mathbf{x}_{-i})u_i(\mathbf{x})f_i'(x_i)}{f_i(x_i)}dF(\mathbf{x})\right)^2 \leq \int \left(\frac{h(\mathbf{x}_{-i})f_i'(x_i)}{f_i(x_i)}\right)^2 dF(\mathbf{x}) \int (u_i(\mathbf{x})-g_i(x_i))^2 dF(\mathbf{x}).$$

D'après (12), (13), (8) et en intégrant par partie nous avons :

$$\int \frac{f_i'(x_i)u_i(\mathbf{x})}{f_i(x_i)}dF_i(x_i) = -\int \frac{\partial G}{\partial x_i}(\mathbf{x})f_i(x_i)dx_i, \forall \mathbf{x}_{-i}.$$

Nous obtenons alors

$$S_{T_i} - S_i = \frac{\int (u_i(\mathbf{x})-g_i(x_i))^2 dF(\mathbf{x})}{\text{var}(Y)} \geq \text{LBI}(h,i) = \frac{\left(\int \frac{\partial G}{\partial x_i}(\mathbf{x})h(\mathbf{x}_{-i})dF(\mathbf{x})\right)^2}{\text{var}(Y)\int \left(\frac{h(\mathbf{x}_{-i})f_i'(x_i)}{f_i(x_i)}\right)^2 dF(\mathbf{x})}.$$

En calculant une borne de $S_{T_i} - S_i$, nous évitons de prendre en compte l'erreur sur S_i, qui peut être large car S_i est prédominant dans beaucoup d'applications.
Le choix de h n'a pas été considéré. Le premier risque est que $\frac{hf_i'}{f_i}$ soit orthogonal à certains termes g_J. De plus, même en ayant une bonne direction, les produits scalaires avec les g_J peuvent être de signes opposés et engendrer une borne inférieure faible. Une approche itérative semble plus intéressante. Pour $(h_k)_k$ une famille orthogonale de fonctions concordantes avec les hypothèses théoriques, nous avons :

$$S_{T_i} - S_i \geq \sum_k \text{LBI}(h_k, i).$$

L'hypothèse (13) n'est pas vérifiée dans notre cas d'application. Nous n'avons donc pas utilisé cette minoration.

3. Code Mascaret, application et résultats

3.1. Mascaret

Dans le cas d'un écoulement à surface libre, lorsque certaines hypothèses sont vérifiées, les équations de Navier-Stokes pour un fluide peuvent être simplifiées pour obtenir les équations de Saint-Venant à une dimension ([Saint-Venant, 1871]).

Mascaret est un code de calcul industriel développé en Fortran par EDF depuis une vingtaine d'années (voir http://www.opentelemac.org/ et [EDF, 2010] pour plus d'informations). Il permet de résoudre les équations de Saint-Venant dans le cas unidimensionnel. Nous n'avons considéré que le noyau de calcul permanent qui calcule un état hydraulique stationnaire. Plus précisément Mascaret permet de calculer des hauteurs d'eau H en fonction :

- d'une valeur de débit en amont que nous perturberons d'une valeur dQ,
- de données sur la géométrie du lit que nous perturberons en modifiant les dénivellements de quantités dZref,

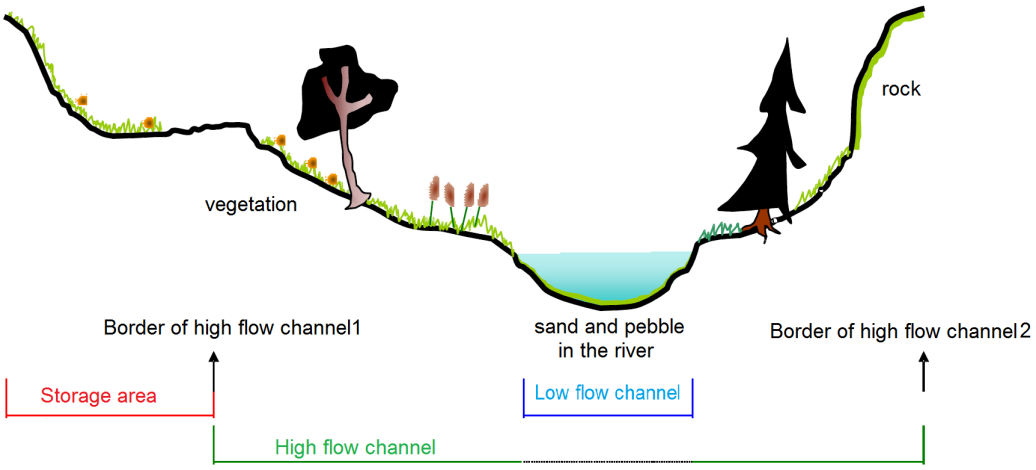


FIGURE 1. ***Schéma d'un profil en travers d'un fleuve. On y distingue le lit mineur (Low flow chanel), et les deux rives du lit majeur (High flow chanel).***

- et de coefficients de frottement de Strickler : cf_2 pour le lit majeur et cf_1 pour le lit mineur (voir le schéma de la figure 1).

Les coefficients de Strickler cf_2 et cf_1 et les perturbations de dénivellement dZref seront dans le cadre de notre étude définis sur les mêmes p zones géographiques.

Nous étudions donc le modèle suivant :

$$H = G(\mathrm{cf1}_1, ..., \mathrm{cf1}_p, \mathrm{cf2}_1, ..., \mathrm{cf2}_\mathrm{p}, \mathrm{dZref}_1, ..., \mathrm{dZref}_p, \mathrm{dQ}).$$

Notons que la résolution par Mascaret des équations de Saint-Venant 1D dans le cas permanent fait intervenir une méthode de la Sécante. Sachant que les opérations intervenant dans cette méthode de calcul sont toutes dérivables et que le nombre d'itérations est assez stable, nous considérerons pour la suite de l'étude que notre modèle est bien dérivable.

3.2. Application

3.2.1. Cas d'étude

Nous avons traité le cas de la Vienne, sur lequel EDF mène des études de crue. Les données d'EDF sont regroupées sur 12 zones d'intérêt, ce qui nous amène à considérer les paramètres physiques suivants, définis par zone :

- 12 coefficients de Strickler correspondant au lit mineur cf1 dont les distributions sont uniformes sur $[20;40]$,
- 12 coefficients de Strickler correspondant au lit majeur cf2 dont les distributions sont uniformes sur $[10;30]$,

- 12 perturbations de dénivellement dZref dont les distributions sont normales standard tronquées sur $[-3;3]$,
- et une valeur de perturbation de débit en amont dQ dont la distribution est gaussienne centrée, d'écart-type 50 et tronquée sur $[-150;150]$.

Les lois de ces paramètres sont choisies arbitrairement, en fonction des distributions empiriques constatées lors de campagnes de mesures. Nous supposons en outre que ces variables aléatoires sont indépendantes, ce qui nous place dans le cadre de la méthodologie présentée précédemment.

En sortie nous prendrons une hauteur d'eau calculée sur une section de calcul d'intérêt, située dans la onzième zone géographique.

Pour de telles entrées aléatoires, une simulation de Mascaret s'exécute en moyenne en $0.20s$.

3.2.2. Obtention et évaluation de l'adjoint

Comme précisé dans la première section de cette article, nous avons utilisé le logiciel Tapenade, développé par l'Inria (pour plus d'information, voir http://www-sop.inria.fr/tropics/tapenade.html).

Évaluation numérique
Nous avons souhaité évaluer notre mode adjoint en le comparant aux différences finies. Nous avons fixé un pas numérique $h = 10^{-5}$ et avons comparé les valeurs de dérivées obtenues à l'aide d'un schéma avant d'ordre 1 à celles obtenues par notre adjoint. Par exemple pour $\mathrm{cf1}_1$ nous comparons la dérivée obtenue par différentiation automatique à :

$$\frac{G(\mathrm{cf1}_1 + h, ..., \mathrm{dQ}) - G(\mathrm{cf1}_1, ..., \mathrm{dQ})}{h}.$$

Nous avons effectué cette comparaison sur l'ensemble des variables sur un tirage de *Monte-Carlo* de taille 10^4.
Les deux méthodes donnent des résultats identiquement nuls pour 32 de nos variables. Aucune valeur n'est nulle en dehors de ces 32 variables et nous représentons la distribution des erreurs relatives des valeurs restantes sur la figure 2. L'erreur relative est toujours inférieure ou égale à 10^{-4}.
Pour une fonction de g de classe C^2, un développement de Taylor à l'ordre 2 en x donne, si $g'(x) \neq 0$:

$$\frac{\frac{g(x+h)-g(x)}{h} - g'(x)}{g'(x)} = \frac{hg"(x)}{2g'(x)} + O(h^2).$$

Le fait que les erreurs relatives soient d'ordres de grandeur voisins de celui de h est un bon indicateur de la validité de notre adjoint.

Coût
Le gain en temps d'exécution est très conséquent, le mode adjoint s'exécute en $0.6s$, ce qui est 3 fois plus lent que le modèle initial et retourne un gradient en une simulation. Notons que ce facteur sort de la fourchette évoquée dans la littérature (voir [Griewank and Walther, 2008], l'adjoint

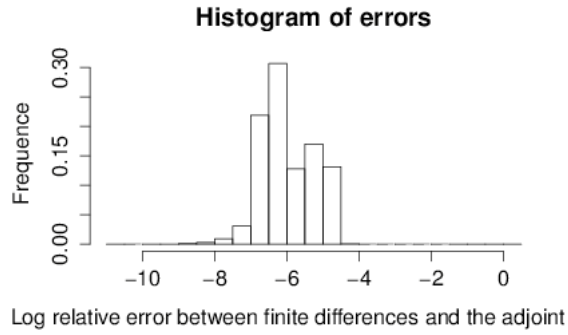


FIGURE 2. ***Logarithme décimal de l'erreur relative entre l'adjoint et les différences finies pour un pas*** $h = 10^{-5}$.

est en général 4 à 6 fois plus lent). Il est possible que ce soit dû à un coût fixe temporel dans l'implémentation (pré et post-traitement).
A titre de comparaison, évaluer un gradient par différences finies en dimension 37 nécessite 38 appels à Mascaret, ce qui s'exécute en moyenne en 7.6*s*. L'utilisation de l'adjoint permet donc ici de gagner un facteur 12.

3.2.3. Valeurs de références

Nous prendrons comme valeurs de référence les résultats obtenus par la méthode de type *Monte-Carlo* ([Saltelli, 2002]) avec une taille d'échantillon $n = 10^5$. Nous assortissons ces valeurs à des intervalles de confiance de niveau 95% obtenus par un bootstrap ([Davison and Hinkley, 1997]) de 2000 répétitions.
Nous constatons que 32 variables ont un indice total estimé comme nul et nous recensons les résultats obtenus pour les 5 variables restantes dans le tableau 2.
Nous ne sommes pas en présence de fortes interactions. En effet, certains intervalles de confiance sont disjoints, mais les écarts numériques entres les indices totaux et du premier ordre restent très faibles.

Le lit majeur n'est pas susceptible d'être inondé dans notre cas d'étude. Il est donc tout à fait cohérent que les coefficients de frottement s'y rapportant n'aient pas d'influence. De plus, en rappelant que la hauteur d'eau que nous considérons en sortie se situe dans la onzième zone géographique, on remarque que les données de frottement et de dénivellement situés en amont de cette zone n'ont pas d'influence et que celles situées en aval n'en ont que très peu. Seuls le débit en amont, le dénivellement et le coefficient de frottement du lit mineur de la zone considérée contribuent réellement à la variance.

3.2.4. Procédure d'estimation

Nous avons d'une part exécuté la procédure de type *Monte-Carlo* ([Saltelli, 2002]) pour $n = 2 * 10^4$ ($(37+2) * 2 * 10^4$) calculs).
D'autre part, pour comparer la méthode de type *Monte-Carlo* à l'approche combinant la méthode

	$cf1_{11}$	$cf1_{12}$	$dZref_{11}$
S	0.4478	0.008529	0.2875
S_T	0.4558	0.01591	0.2928
$IC_{0.95}(S)$	[0.4430;0.4525]	[0.002037 ; 0.01447]	[0.2817 ; 0.2933]
$IC_{0.95}(S_T)$	[0.4517 ; 0.4601]	[0.01564 ; 0.01618]	[0.2898 ; 0.2960]

	$dZref_{12}$	Q
S	0.008813	0.2286
S_T	0.01507	0.2387
$IC_{0.95}(S)$	[0.002252 ; 0.01467]	[0.2230 ; 0.2344]
$IC_{0.95}(S_T)$	[0.01478 ; 0.01538]	[0.2362 ; 0.2411]

TABLE 2. ***Indices totaux et du premier ordre obtenus sur un échantillon Monte-Carlo de taille*** $n = 10^5$ ***avec des intervalles de confiance bootstrap de niveau*** 95***%.***

de lissage et les DGSM, nous avons généré deux échantillons i.i.d. de taille $n = 2 * 10^4$ et calculé les sorties de Mascaret et du mode Adjoint :

- $(\mathbf{x}^k, G(\mathbf{x}^k))_{1\leq k\leq n}$,
- et $(\tilde{\mathbf{x}}^k, \text{grad}_G(\tilde{\mathbf{x}}^k))_{1\leq k\leq n}$.

Nous évaluons ensuite les quantités d'intérêt comme suit :

- Les indices du premier ordre sont estimés via la méthode de lissage (voir [Da Veiga et al., 2009]), appliquée à l'échantillon $(\mathbf{x}^k, G(\mathbf{x}^k))_{1\leq k\leq n}$. Nous notons S_{reg} l'estimateur obtenu. Cette procédure implique l'estimation pour chaque variable i de la fonction $x \mapsto \mathbb{E}(G(\mathbf{X})|X_i = x)$ par un méta-modèle obtenu sur l'échantillon $(x_i^k, G(\mathbf{x}^k))_{1\leq k\leq n}$.
- Pour les espérances conditionnelles aux bords, nous avons utilisé pour la variable i les prédictions sur $\min(x_i^k, 1 \leq k \leq n)$ et $\max(x_i^k, 1 \leq k \leq n)$ du méta-modèle évoqué ci-dessus.
- Les moments de G et $\frac{\partial G}{\partial x_i}$ ont été estimés par des moyennes empiriques classiques.
- Les constantes c et C pour SK_{Inf} et SK_{Sup} ont été estimées respectivement par $\min(\frac{\partial G}{\partial x_i}(\tilde{\mathbf{x}}^k), 1 \leq k \leq n)$ et $\max(\frac{\partial G}{\partial x_i}(\tilde{\mathbf{x}}^k), 1 \leq k \leq n)$.

Pour présenter nos résultats nous confondons les noms des indices mathématiques avec ceux des estimateurs. Nous présentons donc :

- S et S_T obtenus par *Monte-Carlo*,
- S_{reg} obtenu par lissage,
- les estimateurs de $\text{UB}_1, \text{SK}_{\text{Inf}}, \text{SK}_{\text{Sup}}, \text{UB}_2$ (voir 1),
- les estimateurs de LB_3 dans le cas uniforme et de LB_4 dans le cas gaussien tronqué que nous noterons sans distinction LB,
- et S^{ref} et S_T^{ref}, les valeurs de références.

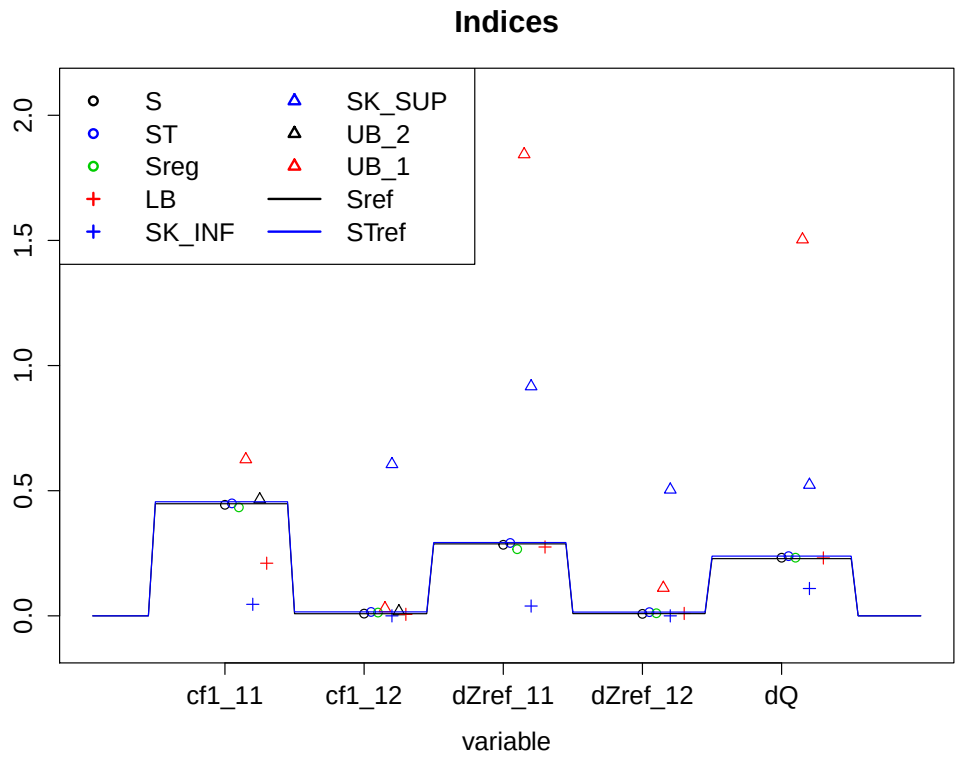


FIGURE 3. ***Les différents estimateurs pour*** $n = 20000$ ***sont comparés aux valeurs de référence (en traits pleins) pour les 5 variables non criblées. Notons que sur ces graphiques les valeurs de référence*** S^{ref} ***et*** S_T^{ref} ***sont quasiment confondues. L'indice*** SK_{Sup} ***n'est ici pas visible pour la variable*** cf1_{11} ***: son estimateur vaut approximativement*** 2.08.

3.2.5. Résultats globaux

La méthode de type *Monte-Carlo* fut exécutée en 44 heures tandis que la procédure alliant la méthode de lissage aux DGSM fut exécutée en 4 heures et 40 minutes. Les calculs n'ont pas été parallélisés mais cela ne favorise aucune des deux approches car elles sont toutes les deux parfaitement parallélisables.

Comme attendu les bornes supérieures nous permettent effectivement de cribler les variables non-influentes. Nous présentons les résultats pour la totalité des variables restantes sur la figure 3 ainsi que pour quelques exemples de variables criblées sur la figure 4.
Les ordres de comparaison sont bien respectés. Au delà du criblage on remarque dans le cas uniforme que la comparaison de UB_2 avec S_{reg} permet de détecter l'absence d'interaction (voir pour $cf1_{11}$ et $cf1_{12}$). Malheureusement les très faibles interactions sur ce cas d'application ne permettent pas d'observer les défauts évoqués pour LB. On remarque néanmoins que cette borne inférieure est une bonne approximation de l'indice du premier ordre dans le cas gaussien tronqué, même si l'intérêt est limité dans notre cas d'application. Notons que SK_{Inf} est souvent nulle car l'existence d'un zéro de $\frac{\partial G}{\partial x_i}$ suffit pour l'annuler.

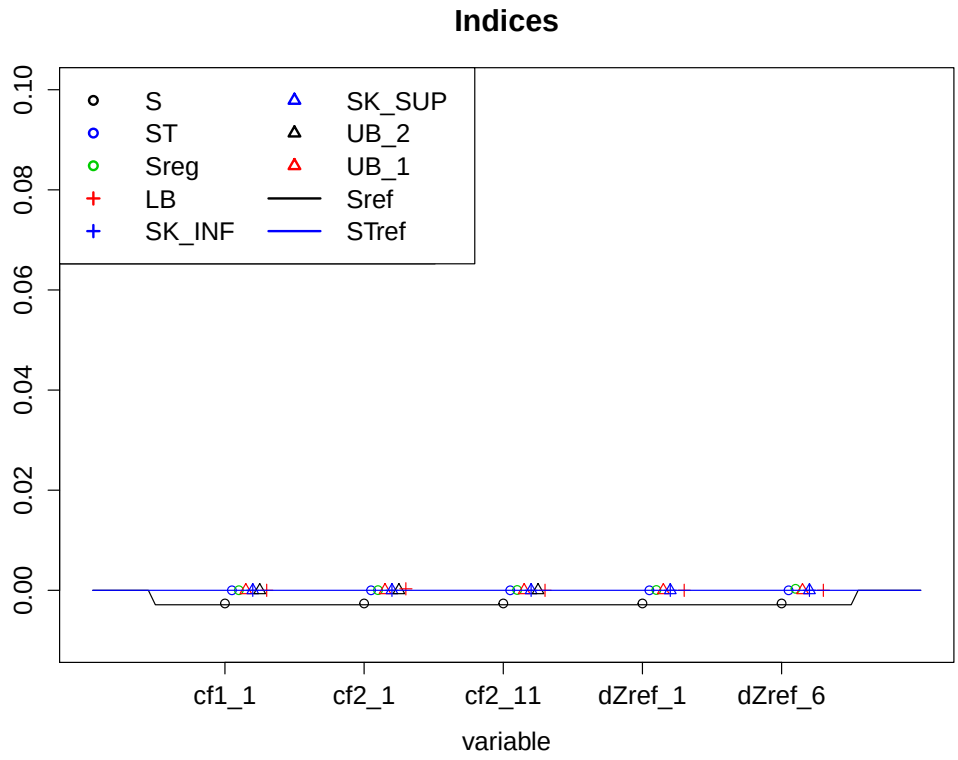


FIGURE 4. ***Les différents estimateurs pour*** $n = 20000$ ***sont comparés aux valeurs de référence (en traits pleins). Nous ne présentons les résultats que pour quelques-unes des variables criblées. Tous les estimateurs sont nuls à l'exception de*** LB ***dont l'ordre de grandeur est*** 10^{-5} ***et des estimations par Monte-Carlo de l'indice du premier ordre (l'estimateur affiche une mauvaise précision même pour*** $n = 10^5$***).***

3.2.6. Comportement des estimateurs

Nous avons exécuté précédemment :

- la procédure combinant la méthode de lissage et le calcul des DGSM en n appels au code et n appels à son adjoint,
- et la méthode de type *Monte-Carlo* en $(d+2)n$ appels au code,

où $n = 20000$.

Nous avons fait varier n et étudié le comportement de ces estimateurs.

Tout d'abord nous remarquons que toutes les bornes supérieures sont nulles pour toute taille d'échantillon pour toute variable non-influente. L'avantage des DGSM pour le criblage par rapport à une méthode de Morris dont le coût augmente linéairement avec d est donc indéniable.

De plus nous représentons pour les variables $\mathrm{cf1}_{11}$ et dZref_{12} les valeurs des différents estimateurs en fonction de n sur les figures 5 et 6. On remarque que les estimateurs de UB_1, UB_2 et LB se stabilisent globalement plus vite en fonction de n que ceux de *Monte-Carlo*.
Les bornes $\mathrm{SK}_{\mathrm{Inf}}$ et $\mathrm{SK}_{\mathrm{Sup}}$ ne fournissent pas de résultat intéressant sur cet exemple. On constate, notamment pour la borne supérieure, que la convergence est assez lente. L'estimateur varie par à-coups et nous ne sommes pas assurés de la convergence pour $n = 20000$. Ce comportement était prévisible pour une recherche d'extremums en dimension 37 par une méthode de *Monte-Carlo*.

4. Conclusion

Pour conclure nous souhaitons rappeler les points suivants.

- Le calcul via l'adjoint des bornes supérieures basées sur les dérivées nous a permis de cribler, c'est-à-dire d'éliminer les variables non influentes, lors de notre étude de manière très rapide. Nous avons aussi pu détecter une absence d'interaction en un temps réduit dans le cas uniforme.
- Le noyau permanent du code Mascaret a été correctement différentié, ce qui rend possible l'étude d'une large catégorie de cas Mascaret permanents via cette méthode.
- Nous avons exhibé des limites théoriques pour la détection d'interactions à l'aide de certaines bornes inférieures. Nous avons proposé une piste d'amélioration.

La suite de ces travaux consistera à procéder de même pour le code Telemac
(voir http://www.opentelemac.org/) et à creuser la piste proposée pour la détection d'interactions. Le code Telemac permet de traiter les écoulements à surface libre en 2D (Saint-Venant) et 3D (Navier-Stokes). Le coût de la procédure est donc plus critique et l'obtention de l'adjoint sera également plus difficile : une approche par surcharge d'opérateurs est envisagée.

Remerciements

Nous souhaitons remercier l'Inria et souligner la grande efficacité du logiciel Tapenade. En particulier nous remercions Valerie Pascual (chercheur INRIA, équipe projet ECUADOR, Sophia-Antipolis) qui nous a grandement aidé pour l'obtention de l'adjoint. Nous tenons également à

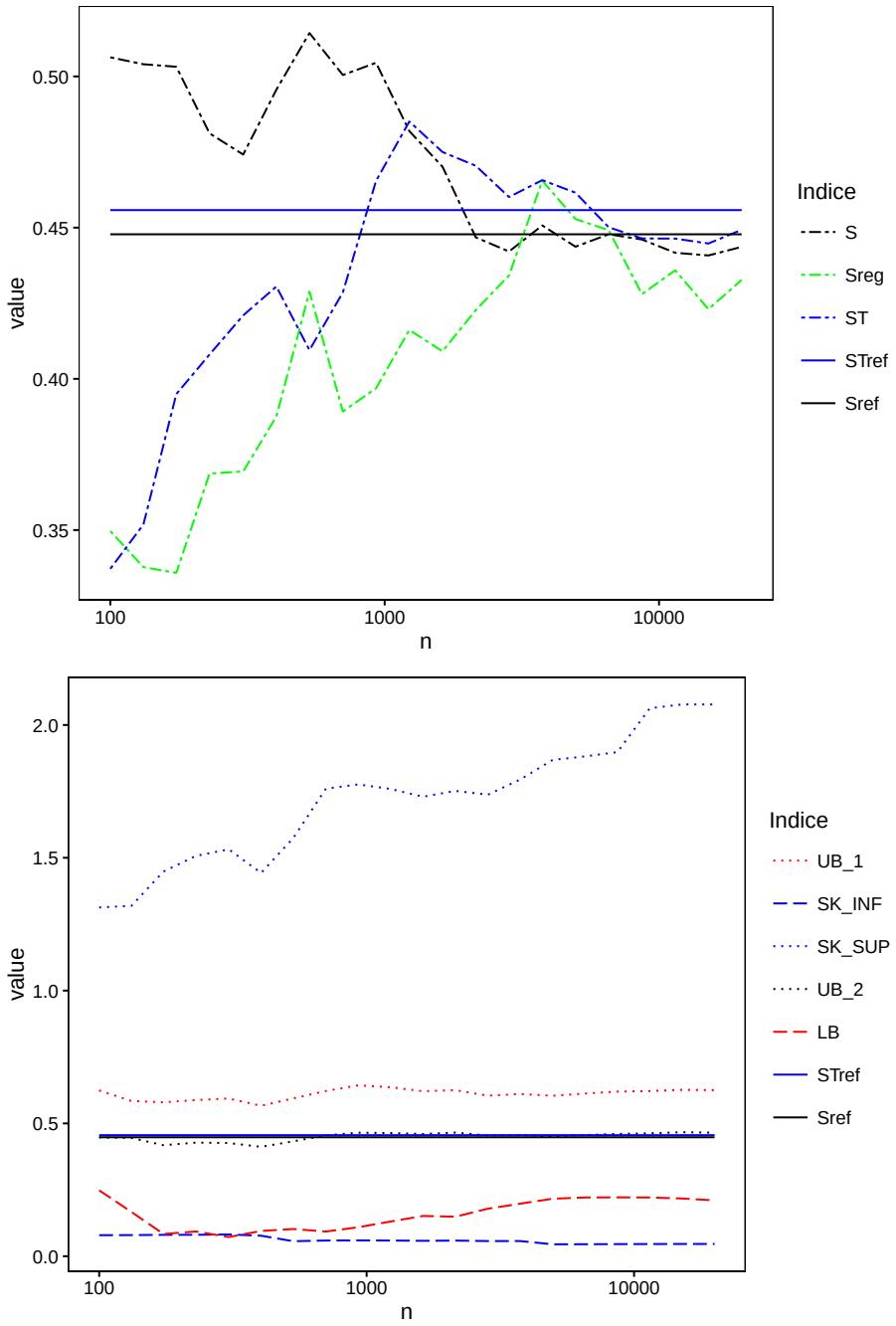


FIGURE 5. ***Évolution des différentes valeurs en fonction de*** n ***pour la variable*** $\mathrm{cf1}_{11}$. ***Les valeurs de référence*** S^{ref} ***et*** S_T^{ref} ***apparaissent ici confondues.***

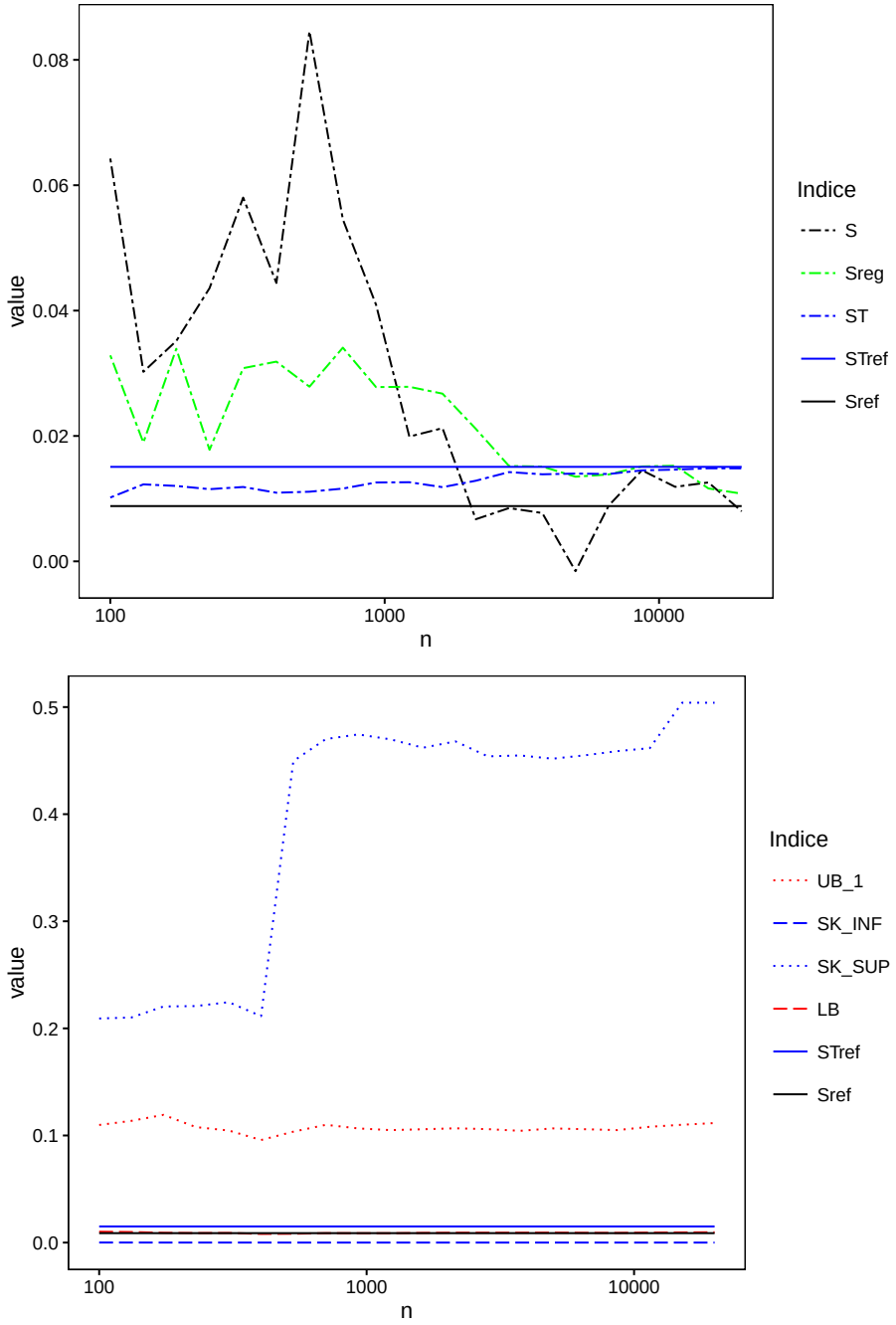


FIGURE 6. ***Évolution des différentes valeurs en fonction de*** n ***pour la variable*** dZref_{12}***. La valeur de*** LB ***reste très proche de*** S^{ref}

remercier Bertrand Iooss (Ingénieur Chercheur Sénior, département MRI, EDF Lab Chatou) pour son expertise sur les méthodes d'analyses de sensibilité.

References

[Da Veiga et al., 2009] Da Veiga, S., Wahl, F., and Gamboa, F. (2009). Local polynomial estimation for sensitivity analysis on models with correlated inputs. *Technometrics*, 51(4):452–463.

[Davison and Hinkley, 1997] Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press.

[EDF, 2010] EDF (2010). *MASCARET v8.1 : Note de principe*.

[Efron and Stein, 1981] Efron, B. and Stein, C. (1981). The jacknife estimate of variance. *The Annals of Statistics*, 9:586–596.

[Fan and Gijbels, 1992] Fan, J. and Gijbels, I. (1992). Variable bandwidth and local linear regression smoothers. *The Annals of Statistics*, 20(4):2008–2036.

[Fan et al., 1996] Fan, J., Gijbels, I., Hu, T. C., and Huang, L. S. (1996). An asymptotic study of variable bandwidth selection for local polynomial regression. *Statistica Sinica 6*, pages 113–127.

[Griewank and Walther, 2008] Griewank, A. and Walther, A. (2008). *Evaluating Derivatives*.

[Iooss and Lemaître, 2015] Iooss, B. and Lemaître, P. (2015). A review on global sensitivity analysis methods. In Meloni, C. and G. Dellino (eds), S., editors, *Uncertainty Management in Simulation-Optimization of Complex Systems: Algorithms and Applications*.

[Kucherenko and Iooss, 2015] Kucherenko, S. and Iooss, B. (2015). Derivative based global sensitivity measures.

[Kucherenko and Song, 2015] Kucherenko, S. and Song, S. (2015). Derivative-based global sensitivity measures and their link with Sobol' sensitivity indices. In Cools, R. and Nuyens, D., editors, *Proceedings of the Eleventh International Conference on Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing (MCQMC 2014)*, Leuven, Belgium. Springer-Verlag,.

[Lamboni et al., 2013] Lamboni, M., Iooss, B., Popelin, A.-L., and Gamboa, F. (2013). Derivative-based global sensitivity measures: general links with sobol' indices and numerical tests. *Mathematics and Computers in Simulation*, 87:45–54.

[Saint-Venant, 1871] Saint-Venant, A. (1871). Théorie du mouvement non-permanent des eaux, avec application aux crues des rivières et à l'introduction des marées dans leur lit. *Compte-rendu à l'académie des Sciences de Paris, 73, 147-154*.

[Saltelli, 2002] Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer Physics Communication*, 145:280–297.

[Sobol, 1990] Sobol, I. (1990). Sensitivity estimates for non linear mathematical models (in Russian). *Matematicheskoe Modelirovanie*, 2:112–118.

[Sobol and Kucherenko, 2009] Sobol, I. and Kucherenko, S. (2009). Derivative based global sensitivity measures and their links with global sensitivity indices. *Mathematics and Computers in Simulation*, 79:3009–3017.

[Storlie and Helton, 2008] Storlie, C. and Helton, J. (2008). Multiple predictor smoothing methods for sensitivity analysis: Description of techniques. *Reliability Engineering and System Safety*, 93:28–54.