Méta-modélisation et analyse de sensibilité pour les modèles avec sortie spatiale. Application aux modèles de submersion marine.

Tran Vi-Vi Elodie Perrin

N° d'ordre NNT : 2021LYSEM016

THESE de DOCTORAT DE L'UNIVERSITE DE LYON
opérée au sein de
L'Ecole des Mines de Saint-Etienne

Ecole Doctorale N° 488
Sciences, Ingénierie, Santé

Spécialité de doctorat : Mathématiques Appliquées
Discipline : Science des Données

Soutenue publiquement le 12/05/2021, par :

Tran Vi-vi Élodie PERRIN

Méta-modélisation et analyse de sensibilité pour les modèles avec sortie spatiale. Application aux modèles de submersion marine.

Metamodelling and sensitivity analysis for models with spatial output. Application to coastal flooding models.

Devant le jury composé de :

Robert FAIVRE	*Directeur de recherche, INRAE*	*Président*
Céline HELBERT	*Maître de conférences, École Centrale Lyon*	*Rapportrice*
Hervé MONOD	*Directeur de recherche, INRAE*	*Rapporteur*
Sophie RICCI	*Chercheur senior, Cerfacs*	*Examinatrice*
Olivier ROUSTANT	*Professeur, INSA Toulouse*	*Directeur de thèse*
Mireille BATTON-HUBERT	*Professeur, Mines Saint-Etienne*	*Directrice de thèse*
Jérémy ROHMER	*Ingénieur de recherche, BRGM Orléans*	*Co-encadrant*
Jean-Philippe NAULIN	*Ingénieur d'étude, CCR Paris*	*Co-encadrant*

Spécialités doctorales	Responsables :	Spécialités doctorales	Responsables
SCIENCES ET GENIE DES MATERIAUX	K. Wolski Directeur de recherche	MATHEMATIQUES APPLIQUEES	O. Roustant, Maître-assistant
MECANIQUE ET INGENIERIE	S. Drapier, professeur	INFORMATIQUE	O. Boissier, Professeur
GENIE DES PROCEDES	F. Gruy, Maître de recherche	SCIENCES DES IMAGES ET DES FORMES	JC. Pinoli, Professeur
SCIENCES DE LA TERRE	B. Guy, Directeur de recherche	GENIE INDUSTRIEL	N. Absi, Maitre de recherche
SCIENCES ET GENIE DE L'ENVIRONNEMENT	D. Graillot, Directeur de recherche	MICROELECTRONIQUE	Ph. Lalevée, Professeur

EMSE : Enseignants-chercheurs et chercheurs autorisés à diriger des thèses de doctorat (titulaires d'un doctorat d'État ou d'une HDR)

ABSI	Nabil	MR	Génie industriel	CMP
AUGUSTO	Vincent	MR	Génie industriel	CIS
AVRIL	Stéphane	PR	Mécanique et ingénierie	CIS
BADEL	Pierre	MA(MDC)	Mécanique et ingénierie	CIS
BALBO	Flavien	PR	Informatique	FAYOL
BASSEREAU	Jean-François	PR	Sciences et génie des matériaux	SMS
BATTON-HUBERT	Mireille	PR	Sciences et génie de l'environnement	FAYOL
BEIGBEDER	Michel	MA(MDC)	Informatique	FAYOL
BLAYAC	Sylvain	MA(MDC)	Microélectronique	CMP
BOISSIER	Olivier	PR	Informatique	FAYOL
BONNEFOY	Olivier	PR	Génie des Procédés	SPIN
BORBELY	Andras	MR(DR2)	Sciences et génie des matériaux	SMS
BOUCHER	Xavier	PR	Génie Industriel	FAYOL
BRODHAG	Christian	DR	Sciences et génie de l'environnement	FAYOL
BRUCHON	Julien	MA(MDC)	Mécanique et ingénierie	SMS
CAMEIRAO	Ana	MA(MDC)	Génie des Procédés	SPIN
CHRISTIEN	Frédéric	PR	Science et génie des matériaux	SMS
DAUZERE-PERES	Stéphane	PR	Génie Industriel	CMP
DEBAYLE	Johan	MR	Sciences des Images et des Formes	SPIN
DEGEORGE	Jean-Michel	MA(MDC)	Génie industriel	Fayol
DELAFOSSE	David	PR0	Sciences et génie des matériaux	SMS
DELORME	Xavier	MA(MDC)	Génie industriel	FAYOL
DESRAYAUD	Christophe	PR	Mécanique et ingénierie	SMS
DJENIZIAN	Thierry	PR	Science et génie des matériaux	CMP
BERGER-DOUCE	Sandrine	PR	Sciences de gestion	FAYOL
DRAPIER	Sylvain	PR	Mécanique et ingénierie	SMS
DUTERTRE	Jean-Max	MA(MDC)		CMP
EL MRABET	Nadia	MA(MDC)		CMP
FAUCHEU	Jenny	MA(MDC)	Sciences et génie des matériaux	SMS
FAVERGEON	Loïc	CR	Génie des Procédés	SPIN
FEILLET	Dominique	PR	Génie Industriel	CMP
FOREST	Valérie	MA(MDC)	Génie des Procédés	CIS
FRACZKIEWICZ	Anna	DR	Sciences et génie des matériaux	SMS
GARCIA	Daniel	MR(DR2)	Sciences de la Terre	SPIN
GAVET	Yann	MA(MDC)	Sciences des Images et des Formes	SPIN
GERINGER	Jean	MA(MDC)	Sciences et génie des matériaux	CIS
GOEURIOT	Dominique	DR	Sciences et génie des matériaux	SMS
GONDRAN	Natacha	MA(MDC)	Sciences et génie de l'environnement	FAYOL
GONZALEZ FELIU	Jesus	MA(MDC)	Sciences économiques	FAYOL
GRAILLOT	Didier	DR	Sciences et génie de l'environnement	SPIN
GROSSEAU	Philippe	DR	Génie des Procédés	SPIN
GRUY	Frédéric	PR	Génie des Procédés	SPIN
HAN	Woo-Suck	MR	Mécanique et ingénierie	SMS
HERRI	Jean Michel	PR	Génie des Procédés	SPIN
KERMOUCHE	Guillaume	PR	Mécanique et Ingénierie	SMS
KLOCKER	Helmut	DR	Sciences et génie des matériaux	SMS
LAFOREST	Valérie	MR(DR2)	Sciences et génie de l'environnement	FAYOL
LERICHE	Rodolphe	CR	Mécanique et ingénierie	FAYOL
MALLIARAS	Georges	PR	Microélectronique	CMP
MOLIMARD	Jérôme	PR	Mécanique et ingénierie	CIS
MOUTTE	Jacques	CR	Génie des Procédés	SPIN
NAVARRO	Laurent	CR		CIS
NEUBERT	Gilles			FAYOL
NIKOLOVSKI	Jean-Pierre	Ingénieur de recherche	Mécanique et ingénierie	CMP
NORTIER	Patrice	PR	Génie des Procédés	SPIN
O CONNOR	Rodney Philip	MA(MDC)	Microélectronique	CMP
PICARD	Gauthier	MA(MDC)	Informatique	FAYOL
PINOLI	Jean Charles	PR	Sciences des Images et des Formes	SPIN
POURCHEZ	Jérémy	MR	Génie des Procédés	CIS
ROUSSY	Agnès	MA(MDC)	Microélectronique	CMP
SANAUR	Sébastien	MA(MDC)	Microélectronique	CMP
SERRIS	Eric	IRD		FAYOL
STOLARZ	Jacques	CR	Sciences et génie des matériaux	SMS
TRIA	Assia	Ingénieur de recherche	Microélectronique	CMP
VALDIVIESO	François	PR	Sciences et génie des matériaux	SMS
VIRICELLE	Jean Paul	DR	Génie des Procédés	SPIN
WOLSKI	Krzystof	DR	Sciences et génie des matériaux	SMS
XIE	Xiaolan	PR	Génie industriel	CIS
YUGMA	Gallian	CR	Génie industriel	CMP

Mise à jour : 09/11/2020

Remerciements

Il me sera très difficile de remercier tout le monde car de nombreuse personnes ont contribué à l'aboutissement de cette thèse. Néanmoins, je souhaite mettre en avant certaines d'entre elles.

Je voudrais avant tout remercier Olivier ROUSTANT, qui m'a encadré tout au long de cette thèse. Ses précieux conseils m'ont permis d'évoluer scientifiquement, professionnellement, humainement, et m'ont aidé à devenir plus rigoureuse et minutieuse dans mon travail.

Je remercie aussi mes co-encadrants BRGM et CCR, Jérémy ROHMER et Jean-Philippe NAULIN, pour leurs disponibilités et leurs aides précieuses, même à distance.

Je tiens également à remercier Mireille BATTON-HUBERT, responsable du département Génie mathématique et industriel (GMI) à l'Institut Fayol, qui m'a accueillie dans son équipe, et qui a apporté l'aide nécessaire face à des problèmes délicats qui auraient pu mettre fin à ma thèse dès la première année.

Mes remerciements sont de plus adressés à Céline HELBERT, maître de conférences à l'École Centrale de Lyon, et Hervé MONOD, directeur de recherche à l'INRAE, d'avoir accepté d'être rapporteurs de cette thèse. Je remercie de même Robert FAIVRE, directeur de recherche à l'INRAE, et Sophie RICCI, chercheur sénior à Cerfacs, qui ont bien voulu être examinateurs.

J'exprime ma gratitude à Olivier ALATA, professeur à l'Université Jean Monnet, d'avoir consacré du temps dans le partage de ses connaissances de la théorie des ondelettes, et l'écriture d'un article, publié dans la revue *Reliability Engineering & System Safety*.

Je remercie aussi tous les collègues rencontrés au BRGM et à la CCR, qui m'ont bien accueillie dans leurs établissements à chaque déplacement. Les échanges effectués en réunion ou à la machine à café m'ont permis de découvrir davantage l'évaluation des risques de catastrophes naturelles (autres que la submersion marine) et la réassurance

(CCR).

Je remercie en particulier David MONCOULON et Pierre TINARD de la CCR, pour leurs intérêts aux recherches de la thèse et pour les différentes discussions qui m'ont permis de proposer des axes d'améliorations. Mes remerciements sont aussi pour Rodrigo PEDREROS et Déborah IDIER du BRGM, pour avoir partagé avec moi leurs expertises.

La thèse a été majoritairement réalisée à l'École des Mines de Saint-Étienne (EMSE), dans l'équipe GMI, à l'Institut Fayol. Je tiens donc à remercier tous les collègues de l'équipe GMI, qui m'ont aidée et soutenue durant la thèse. Je garderai de bons souvenirs des pauses café et thé, où les échanges étaient très animés, accompagnés parfois de pâtisseries préparées par mes soins.

Je remercie notamment mon co-bureau Vincent (pour tous nos échanges sur le Machine Learning et tes concerts de trompette quotidiens), Andres (pour avoir répondu à mes questions quotidiennes, expliqué le développement d'un package sous **R**, et aidé à me perfectionner en LaTeX), Audrey (pour tous les moments conviviaux passés ensemble et les pauses café virtuelles), David (pour m'avoir appris les bases du git, et d'avoir partagé tes bons de réduction du supermarché), Nicolas (pour m'avoir fait découvrir tous les secrets des processus gaussiens et des jeux de société), Sawssen (c'est toujours un plaisir de discuter avec toi), et enfin Marie DELLISE pour son très grand soutien lors des épreuves difficiles.

Mes remerciements s'adressent à Matthieu BALLAIRE qui m'a soutenue pendant la rédaction du manuscrit. Enfin, je remercie ma famille et mes proches, plus particulièrement mes parents, Aï-Nhi et Jean-Baptiste PERRIN, ma sœur Tu-Van Lyvia, et mon frère, Tran Vu Kévin.

Tran Vi-vi Élodie PERRIN

Table des matières

Table des figures

Liste des tableaux

Notations

On liste ici les notations et abréviations utilisées dans les parties II, III, et IV.

PG	Processus Gaussien
AS	Analyse de Sensibilité
ACP	Analyse en Composantes Principales
ACPF	ACP Fonctionnelle
f	le simulateur
d	nombre de paramètres en entrée de f
$\mathcal{X} \subseteq \mathbb{R}^d$	domaine des entrées de f
$\mathbf{x} \in \mathcal{X}$	vecteur des entrées de f
$\mathcal{Z}$	domaine spatial
$D_{\mathbf{z}}$	Si $\mathcal{Z}$ est discrétisé, $D_{\mathbf{z}}$ est la taille de la discrétisation (taille de la grille, nombre de pixels, etc.)
$\mathbf{z} = (z_1, z_2) \in \mathcal{Z}$	couple des coordonnées spatiales
$\mathbb{L}^2(\mathcal{Z})$	espace de Lebesgue du second ordre
$y_{\mathbf{x}} \in \mathbb{L}^2(\mathcal{Z})$	sortie fonctionnelle de f pour le vecteur des entrées $\mathbf{x}$
n	nombre de simulations/observations de f
$\{(\mathbf{x}_i, y_i(\mathbf{z})), i = 1, \ldots, n\}$	l'ensemble des observations de f
$\mathbf{x}^\star$	le vecteur d'entrées où prédire f à partir des observations $\{(\mathbf{x}_i, y_i(\mathbf{z})), i = 1, \ldots, n\}$
$\Phi(\mathbf{z}) = (\phi_1(\mathbf{z}), \ldots, \phi_K(\mathbf{z}))^\top$	base de fonctions (par exemple, les ondelettes ou le B-splines)
$\boldsymbol{\alpha}(\mathbf{x}) = (\alpha_1(\mathbf{x}), \ldots, \alpha_K(\mathbf{x}))^\top$	vecteurs des coordonnées de $y_{\mathbf{x}}$ dans la base de fonctions $\Phi(\mathbf{z})$
G	matrice de Gram de la base $\Phi(\mathbf{z})$
R	une racine carrée de G, c'est-à-dire telle que $RR^\top = G$
K	dimension de l'espace des sorties ou de la base d'ondelettes
$\tilde{K} \ll K$	nombre de coefficients sélectionnés pour l'ACP
λ_k	critère de sélection des coefficients de la base $\Phi(\mathbf{z})$
$p \in [0, 1]$	paramètre réglant la proportion moyenne de l'énergie
n_{PC}	nombre de composantes principales retenues

$\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{n_PC})$	base des vecteurs propres
$\omega_{(k),l}$	(k)-ème élément du l-ème vecteur propre
$\mathbf{t}(\mathbf{x}) = (t_1(\mathbf{x}), \ldots, t_{n_PC}(\mathbf{x}))^\top$	vecteur des coordonnées (scores) dans la base des vecteurs propres
$\lambda(\mathbf{x}) > 0$, avec $\mathbf{x} \in \mathcal{X}$	un paramètre de pénalité
λ	paramètre de pénalité commun quel que soit $\mathbf{x} \in \mathcal{X}$
$\mathrm{GP}^{\mathrm{PCA}}$, $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$, et $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$	les méta-modèles effectués par ACP standard, par ACPF sur une base d'ondelettes, et par ACPF sur une base de B-splines
n_{test}	taille de l'échantillon test
$\xi_1(\mathbf{z}), \ldots, \xi_{n_PC}(\mathbf{z})$	fonctions propres de l'ACPF
GSI_ω	indice de sensibilité spatial généralisé de $y_{\mathbf{x}}(.)$ par rapport à $\mathbf{x}_\omega$, avec $\omega \subseteq \{1, \ldots, d\}$.
T, S, t_0, t_-, t_+	voir le tableau 5.1
$conn_i$, avec $i = 1, \ldots, 7$	nom des variables binaires associées aux connexions hydrauliques
frott.A, ..., frott.L	nom des variables des 12 coefficients de rugosité (voir le tableau 5.5)

Première partie

Introduction

Chapitre 1

Introduction

Sommaire

1.1 Contexte

Les submersions marines font partie des périls majeurs susceptibles d'affecter les régions côtières dans le monde, comme l'ont montré des événements récents tels que le cyclone Irma en 2017 ou l'ouragan Sandy en 2012. En France métropolitaine, le dernier évènement majeur est la tempête Xynthia en 2010, qui a conduit à 53 morts, 79 blessés, et 2.5 milliards d'euros de dommages, dont 700 millions par la submersion marine ([Naulin et al., 2015, FFSA, 2011]). Un élément clef de toute évaluation de risque d'inondation marine est la capacité à prédire de façon précise et robuste ce qu'il pourrait se passer à terre (e.g. les niveaux d'eau, l'étendue spatiale de l'inondation ...) en fonction de ce qu'il se passe au large, i.e. les conditions météorologiques et océaniques comme la surcote, l'amplitude de la marée ou les caractéristiques des vagues. Cela peut être fait en utilisant des modèles (simulateurs) hydrodynamiques numériques haute résolution, comme ceux développés par le BRGM (Bureau de Recherches Géologiques et Minières, voir https://www.brgm.fr/fr) et la CCR (Caisse Centrale de Réassurance, voir https://www.ccr.fr/), que l'on appellera les modèles d'aléa. Le BRGM est l'organisme public français de référence dans le domaine des sciences de la Terre pour la gestion des ressources naturelles et des risques du sol et du sous-sol. La CCR est une société française qui propose aux assureurs une couverture de réassurance pour les catastrophes naturelle dans le cadre du régime d'indemnisation des catastrophes naturelles. A ce titre, elle développe des modèles permettant d'anticiper les conséquences financières de ces catastrophes ainsi que de provisionner des fonds. Lorsqu'une catastrophe naturelle survient, la CCR communique une estimation de

l'impact financier d'un événement à ses clients, au grand public et à l'état Français.

Dans la thèse, l'inondation est quantifiée par la profondeur d'eau maximum atteinte (calculée sur la durée d'un événement de tempête donné), qui est la sortie considérée du modèle numérique. Cependant, ces modèles présentent des incertitudes importantes, qui proviennent des données qui les alimentent telles que les forçages marins au large (vagues et niveau d'eau des mers ou des océans) ou l'altitude du modèle numérique de terrain (MNT). D'autres proviennent des paramètres du modèle tels que la rugosité du sol. La thèse, en partenariat entre le BRGM (Bureau de Recherches Géologiques et Minières), la CCR (Caisse Centrale de Réassurance), et l'École des Mines de Saint-Étienne (EMSE), a pour objectif d'identifier et de hiérarchiser les sources d'incertitudes, qui peuvent être améliorées dans les modèles d'aléa.

Les méthodes d'analyse de sensibilité (AS) permettent d'effectuer cette identification et cette hiérarchisation [Iooss, 2011, Iooss and Lemaître, 2015]. Cependant, il y a deux principales problématiques. Premièrement, les méthodes de Monte Carlo, couramment utilisées pour estimer les indices de sensibilité de chaque paramètre d'entrée, nécessitent un grand nombre de simulations. Par conséquent, elles sont difficilement applicables directement sur les simulateurs (entre 30 minutes et 1 heure pour le modèle BRGM, et environ 5 minutes pour la CCR, pour une seule simulation). Deuxièmement, la sortie est fonctionnelle : la profondeur d'eau maximum est une fonction dépendant de la localisation. En pratique, les localisations sont discrétisées sur une grille, et la sortie est représentée par un vecteur de grande dimension, de longueur égale au nombre de pixels. Le niveau de discrétisation requis peut être très fin (jusqu'à quelques mètres). Cela pourrait rajouter des difficultés à l'AS, en imposant de manipuler des vecteurs de grandes dimensions ($K = 256 \times 256 = 65\ 536$, dans le cas d'étude, voir le chapitre 5).

1.2 Généralités sur la submersion marine

Selon [Garry et al., 1999], la submersion marine peut être définie comme « *une inondation temporaire de la zone côtière par la mer dans des conditions météorologiques (forte dépression et vent de mer) et marégraphiques sévères* ». Les submersions marines sont les plus souvent associées à des surélévations temporaires du niveau de la mer lors de tempêtes ou de cyclones.

Les phénomènes de submersion marine résultent généralement de la combinaison entre de forts coefficients de marée et le passage d'une tempête produisant une surcote c'est-à-dire une surélévation du niveau marin. Cette surélévation résulte elle-même de trois facteurs principaux :

— La surcote atmosphérique (ou météorologique) : diminution de la pression atmosphérique qui entraîne une surélévation du niveau de la mer.

— Le wave setup (surcote liée aux vagues) : les vagues générées au large se propagent vers la côte et déferlent sur la colonne d'eau, ce qui provoque une surélévation du plan d'eau.
— Le vent : les forces de frottement du vent peuvent modifier les courants, lever les vagues, et accumuler l'eau sur la bande littorale.

Le niveau d'eau en mer (ou de l'océan) résulte de la contribution de la surcote atmosphérique, du wave setup et de la marée. Enfin, il dépendra du jet de rive (swash), c'est-à-dire le flux et le reflux des vagues sur l'estran (partie du littoral périodiquement recouverte par la marée). Le runup correspond à la cote maximale atteinte par la mer au-dessus du niveau « normal » de la marée.

On peut distinguer trois types de submersion marine (illustrés dans la figure 1.1) :
— le **franchissement** par paquets de mer : passage des vagues par-dessus les défenses côtières (naturelles ou artificielles).
— le **débordement** : élévation du niveau d'eau au-dessus de la cote maximale du terrain ou des ouvrages de protections. Cela entraîne un déversement direct des eaux à terre. Les volumes d'eau déversés dépendent de l'écart entre le niveau d'eau et celui de la cote maximale.
— la **rupture** de structures de protection, à la suite de dégradations progressives causées par les vagues.

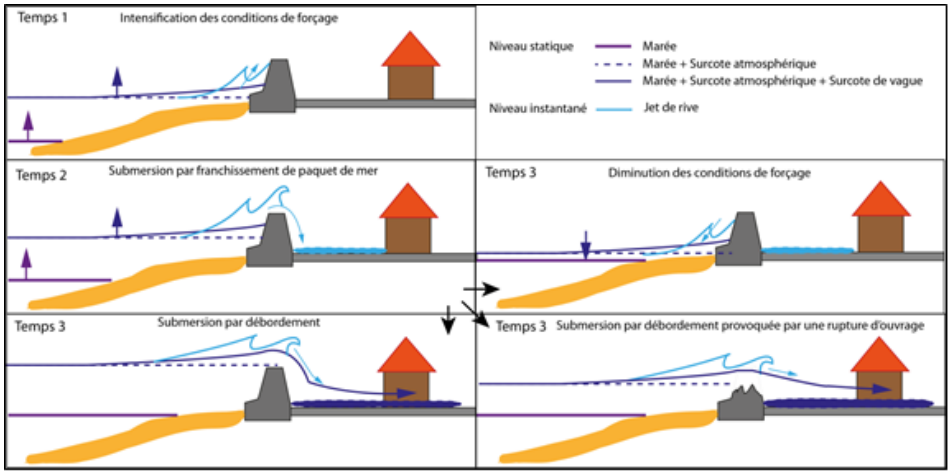

FIGURE 1.1 – Exemple de processus entraînant une submersion marine (franchissement par paquets de mer, rupture des structures de protection/débordement, ©BRGM).

Une submersion marine peut résulter de l'un ou de plusieurs de ces trois processus. Ils peuvent aussi se produire simultanément en des endroits différents le long du linéaire côtier.

1.3 Données nécessaires à la modélisation

Le forçage marin (niveau de la marée, surcote, etc.) est nécessaire à la modélisation de la submersion marine. En pratique, il peut être lui-même estimé par des modèles hydrodynamiques (Telemac, Previmer, etc.), et est donc source d'incertitudes pour le modèle d'aléa. D'ailleurs, le BRGM et la CCR utilisent chacun différents forçages. La propagation de la submersion marine est conditionnée par deux autres données nécessaires : la topographie du terrain et l'occupation du sol.

Pour l'occupation du sol, en effet, chaque type de sol va plus ou moins présenter une résistance au passage de l'eau. L'impact du sol est représenté via un paramètre de frottement caractéristique de la rugosité du sol (appelé coefficient de rugosité). La rugosité est généralement paramétrée en termes de coefficients de Manning ou de Strickler (inverse du coefficient de Manning). Une liste des coefficients associés à leurs types de sol est donnée dans le tableau 5.5 (illustrée dans la figure 5.23). En pratique, pour chaque type de sol, le coefficient de rugosité est fixé dans un modèle. Ainsi, les modèles de la CCR et du BRGM utilisent la même occupation du sol mais considèrent des coefficients de rugosité différents.

La simulation de la submersion marine nécessite des données d'altitude du terrain (la topographie), afin de bien représenter la géométrie des obstacles (structure de protections côtières, murets, etc.) et des facteurs d'écoulement (canaux, voies d'eau, etc.). La topographie est renseignée dans le modèle d'aléa sous la forme d'un MNT (Modèle Numérique de Terrain), qui est une grille régulière, référencée spatialement, et comportant sur chaque maille une estimation de l'altitude. Il en existe plusieurs en fonction de la zone étudiée, par exemple en France : Intermap (résolution : 25m), BD topo IGN (25m), Litto3D (1m), etc. La résolution de la grille du MNT a son importance. En effet, les estimations de la submersion peuvent être erronées si elles se basent sur des données de résolution spatiale et de précision verticale (altitude) grossières, les obstacles et vecteurs d'écoulement ne pouvant pas être reproduits. En revanche, plus la résolution est fine, plus les temps de calcul sont importants.

Dans le chapitre 5.5, on remarque que l'influence du MNT n'a pas été analysée, contrairement à la rugosité et au forçage marin. En effet, une autre problématique se pose pour le MNT : il s'agit d'une carte qui est en entrée du modèle. Dans la littérature, cette problématique a été traitée, dont [Iooss and Ribatet, 2009] et [Lilburne and Tarantola, 2009] donnent un état de l'art des approches. Dans le travail de thèse, nous nous sommes principalement concentrés sur la problématique des sorties spatialisées car ce sont elles qui sont les plus critiques d'un point de vue opérationnel. Nous avons considéré que le MNT intégré dans les modèles d'aléa : IGN pour la CCR, et Litto3D agrégé à 25m pour le BRGM.

1.4 Analyse des problématiques

Dans ce contexte, une méthodologie standard traite les deux problématiques de la manière suivante (voir [Chen et al., 2011, Marrel et al., 2011, Jia and Taflanidis, 2013, Marrel et al., 2015, Li et al., 2020, Ma et al., 2019]). Premièrement, la dimension de la sortie fonctionnelle (ou spatiale) est réduite, le plus souvent par analyse en composantes principales (ACP) ou en utilisant la décomposition dans une base de fonctions (par exemple, les bases de Fourier ou d'ondelettes). Cela permet de construire un vecteur de sortie de dimension inférieure, formé des plus grandes composantes (coordonnées de l'ACP, aussi appelées scores, ou les coefficients de la base de fonctions). Deuxièmement, une fonction rapide à évaluer, appelée méta-modèle ou modèle de substitution, est construite pour ce vecteur. Cela se fait généralement en considérant indépendamment chaque coordonnée comme une sortie scalaire comme dans [Chen et al., 2011]. Parmi tous les méta-modèles (régression linéaire, réseaux de neurones, etc.), on sélectionne les modèles de régression par processus gaussien (PG) [Rasmussen and Williams, 2006], car ils fournissent une interpolation des données et une incertitude pour des entrées inconnues. De plus, la méthode est paramétrée par une fonction de covariance (ou noyaux), ce qui la rend flexible, et permet d'exploiter la connaissance d'experts.

Cependant, l'ACP traite chaque localisation indépendamment et ne tient pas en compte de la dépendance spatiale. De plus, la sortie du simulateur montre de fortes variations locales (illustrées dans la figure 5.2), la fonction des profondeurs d'eau n'est pas une fonction lisse des localisations. Par conséquent, même avec une base de fonctions adaptée telle que les ondelettes, un grand nombre de coefficients, typiquement plusieurs milliers, doit être gardé afin d'obtenir une approximation précise. Ce problème a été clairement souligné dans des études précédentes (voir [Marrel et al., 2011], [Marrel et al., 2015]) et affaiblit les bénéfices de la réduction de dimension.

Pour pallier le problème, on propose d'utiliser l'ACP fonctionnelle (ACPF), une technique courante de l'analyse de données fonctionnelles [Ramsay and Silverman, 2004]. Elle est équivalente à appliquer une ACP aux coefficients d'une décomposition sur une base fonctionnelle, avec la métrique donnée par la matrice de Gram de la base de fonctions. Des bases de fonctions populaires peuvent être utilisées : Fourier, ondelettes, B-splines. On note que pour les bases non-orthonormées comme les B-splines, l'étape de l'ACP utilise une métrique différente de l'ACP standard. De plus, on ajoute une étape de sélection préliminaire, en choisissant les termes de la base qui ont le plus d'influence, en se basant sur la décomposition de l'énergie (variance spatiale) après l'orthonormalisation de la base, ou selon une approche de régression pénalisée directement sur la base d'origine. Avec ces deux idées, la méthode est applicable pour des vecteurs de grande taille, par exemple, des cartes avec plus de dix mille pixels. De plus, les avantages de l'ACP et de la décomposition sur une base sont cumulés en tenant compte de la dépendance spatiale de la sortie, qui est ignorée par l'ACP standard, puisque les fonctions sont décomposées

dans un espace fonctionnel adapté. En outre, en appliquant l'ACP dans un second temps, la réduction de la dimension est assurée, même quand un grand nombre de coefficients doit être gardé : le nombre final de composantes principales est petit. Finalement, comme remarqué en faisant l'ACP, construire un PG indépendant pour chaque score a autant de sens qu'un modèle de krigeage multivarié, puisque les composantes principales sont décorrélées, bien que non nécessairement indépendantes.

L'utilisation de l'ACPF pour l'AS a été proposée par exemple dans [Lamboni et al., 2011] (voir aussi [Xiao and Li, 2016]) : une base de fonctions orthonormées est obtenue comme les fonctions propres d'un opérateur de Hilbert-Schmidt associé à un noyau de covariance, par décomposition de Karhunen-Loève. Dans notre approche, on définit la base de fonctions en premier. En théorie, les deux approches sont équivalentes, puisqu'un noyau de covariance peut être construit à partir d'une base prédéfinie correspondant à sa décomposition Karhunen-Loève. Cependant, en pratique, ici, il y a un avantage clair à définir une base de fonctions en premier, traitant ainsi la non stationnarité sans la connaissance d'experts. En effet, contrairement aux noyaux RKHS usuels, qui sont guidés par des hypothèses globales de régularité, plusieurs bases fonctionnelles telles que les ondelettes sont adaptées pour approximer des fonctions avec de fortes variations locales.

Dans le cas de séries temporelles, [Campbell et al., 2006] propose d'appliquer l'AS aux coordonnées de l'ACP ou aux coefficients d'une décomposition dans une base fonctionnelle. En analysant la structure des composantes principales ou des fonctions de la base fonctionnelle, on peut interpréter l'influence des entrées sur la structure spatiale de la sortie du modèle. Par application à un cas analytique (voir la section 4.3) et au cas des submersions marines (voir les sections 5.4 et 5.5), on montre l'approche pour les modèles dont la sortie est spatiale (en pratique, représentée par une matrice). Comme autre contribution de la thèse, une expression des indices de Sobol adaptés aux sorties spatiales est développée. La formule est équivalente à l'expression proposée dans [Lamboni et al., 2011] (ces indices sont appelés « indices de sensibilité généralisés ») dans le cas d'utilisation de base orthonormée, quand un noyau est construit à partir d'une base de fonctions. La formule, montrée dans le manuscrit, est aussi valide pour des bases de fonctions non orthonormées populaires telles que les B-splines.

Le travail sur la méta-modélisation de modèles avec sortie spatiale a conduit au développement d'un package **R** (voir le chapitre 6), appelé « GpOutput2D », disponible sur github au lien suivant : https://github.com/tranvivielodie/GpOutput2D.

1.5 Organisation du manuscrit

Pour commencer, le chapitre 2 donne les bases sur les processus gaussiens (PG), l'analyse de sensibilité (AS), notamment sur les indices de Sobol, et l'analyse en composante principale fonctionnelle (ACPF).

Le manuscrit est ensuite organisé en trois parties : II) Présentation de la méthodologie pour la méta-modélisation et l'analyse de sensibilité de modèles avec sortie spatiale, avec une application à un cas analytique ; III) Application aux modèles de submersion marine ; IV) Présentation du package « GpOutput2D », pour la méta-modélisation de modèles avec sortie spatiale.

La partie II présente la méthodologie pour la méta-modélisation et l'analyse de sensibilité de modèles avec sortie spatiale. Elle comporte deux chapitres. Le chapitre 3 décrit la procédure de la méta-modélisation : des modèles de krigeage indépendants sont construits pour chaque score de l'ACPF. Deux méthodes de sélection de coefficients de la base de fonctions choisie (ondelettes ou B-splines), sont expliquées : 1) avec une étape préliminaire d'orthonormalisation, les coefficients sélectionnés correspondant aux coefficients les plus influents selon la décomposition de l'énergie (variance spatiale) ; 2) un modèle de régression pénalisé pour estimer les coefficients de la décomposition, estimant ceux des termes les moins influents à zéros. La méthodologie basée sur l'ACPF, en utilisant les bases d'ondelettes et de B-splines, est comparée à l'approche avec ACP standard, sur un cas analytique. Les deux approches de sélection des coefficients sont aussi comparées. Le chapitre 4 présente la formule équivalente à l'indice de sensibilité généralisé de [Lamboni et al., 2011] dans le cas d'une base non orthonormée. Pour le cas analytique présenté dans le chapitre précédent, on interprète l'AS spatiale de la sortie du modèle via les composantes principales générées par ACPF. Les indices de sensibilité généralisés sont aussi estimés.

La partie III contient le chapitre 5 qui présente les résultats obtenus pour les modèles de submersion marine. En effet, les méthodologies présentées dans la partie II ont été appliquées aux modèles d'aléa du BRGM et de la CCR. La méta-modélisation par ACPF sur bases d'ondelette et de B-splines est comparée à l'approche avec ACP standard. Dans un premier temps, l'influence des paramètres du forçage marin (marée, surcote) est analysée sur l'estimation de l'inondation dans les terres. Ensuite, on se place dans la situation où le modèle d'aléa est utilisé pour prédire les conséquences d'une tempête donnée, ici Xynthia (2010) qui a entraîné une submersion marine aux Bouchôleurs (proche de la Rochelle) en France, pour effectuer l'analyse de sensibilité. Dans ce deuxième cas, les paramètres analysés sont le forçage marin (modèle utilisé), l'ouverture de sept connexions hydrauliques (brèches, etc.) et les coefficients de rugosité. Enfin, ce chapitre se conclut par une étude prospective sur la possibilité d'associer les deux modèles d'aléa par l'ajout d'une variable catégorielle.

La partie IV contient la vignette du package **R** : GpOutput2D. Les implémentations logicielles réalisées contiennent la plupart des développements de la méthodologie présentée dans le chapitre 3. La vignette explique l'utilisation des fonctions principales du package en les appliquant à un cas analytique.

Finalement, dans la partie V, on résume les conclusions de l'ensemble de la thèse. Un aperçu des perspectives est aussi présenté pour plusieurs points : prédiction de cartes lorsqu'il y a inondation ou non, utilisation d'autres mesures d'incertitudes que la variance, détermination d'une loi de distribution pour les composantes principales de l'ACPF, etc.

Chapitre 2

Bases méthodologiques

Sommaire

2.1 Régression par processus gaussien

Ce chapitre s'inspire de [Rasmussen and Williams, 2006] pour les processus gaussiens.

Dans cette section, on s'intéresse à une fonction (ou un simulateur) $f : \mathcal{X} \subseteq \mathbb{R}^d \to \mathbb{R}$, avec $d \geq 1$. On suppose que l'on ignore comment $f(\mathbf{x})$ est calculé pour $\mathbf{x} \in \mathcal{X}$. On peut seulement avoir accès à nombre limité des valeurs de f : $(\mathbf{x}_1, f(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, f(\mathbf{x}_n))$, $n \in \mathbb{N}$. À partir de ces valeurs, on cherche à prédire la valeur de f en un nouveau point

x^*.

2.1.1 Les vecteurs gaussiens

Définition 2.1 (Densité d'une loi gaussienne)**.** *Une variable aléatoire suit une loi gaussienne de moyenne μ et de variance σ^2 si sa fonction de densité de probabilité est :*

$$\phi(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

Remarque 2.1. *Si $N \sim \mathcal{N}(0,1)$ et $X = \mu + \sigma N$ alors $X \sim \mathcal{N}(\mu, \sigma^2)$.*

Définition 2.2 (Vecteur gaussien)**.** *Un vecteur aléatoire $\mathbf{Y} = (Y_1, \ldots, Y_n)^\top$ à valeurs dans $\mathbb{R}^n$ est dit **gaussien** si toute combinaison linéaire de $\mathbf{Y}$ suit une loi gaussienne. C'est à dire :*

$$\forall \mathbf{a} \in \mathbb{R}^n, \exists \mu \in \mathbb{R}, \sigma^2 > 0 \text{ tel que } \mathbf{a}^\top \mathbf{Y} = \sum_{k=1}^{n} a_k Y_k \sim \mathcal{N}(\mu, \sigma^2)$$

.

*Si $\mathbf{Y}$ est un vecteur gaussien, on définit son **vecteur moyenne $\boldsymbol{\mu}$** par*

$$\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n) = (\mathbb{E}(Y_1), \ldots, \mathbb{E}(Y_n))^\top$$

*et sa **matrice de variance-covariance** Σ par*

$$\Sigma = \mathbb{E}((\mathbf{Y} - \mathbb{E}(\mathbf{Y}))(\mathbf{Y} - \mathbb{E}(\mathbf{Y}))^\top)$$

La fonction de densité de probabilité de $\mathbf{Y}$ est donnée par l'équation (2.1).

$$\phi_{\mathbf{Y}}(x) = \frac{1}{|2\pi\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\boldsymbol{\mu})^\top \Sigma^{-1}(x-\boldsymbol{\mu})\right) \tag{2.1}$$

La matrice de variance-covariance Σ est symétrique et semi-définie positive (c'est-à-dire $\forall \alpha \in \mathbb{R}^n,\ \alpha^\top \Sigma \alpha \geq 0$).

Remarque 2.2. *Plus loin, avec $\Sigma = (k(\mathbf{x}_i, \mathbf{x}_j))_{1 \leq i,j \leq n}$, on aura $\alpha^\top \Sigma \alpha = \sum_{i,j} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$, ce qui donnera la définition d'une fonction semi-définie positive.*

Théorème 2.1. *[Théorème de conditionnement gaussien (TCG)] On considère un vecteur gaussien $\mathbf{Y} = (\mathbf{Y}_1, \mathbf{Y}_2)$, où $\mathbf{Y}_1$ et $\mathbf{Y}_2$ sont des vecteurs aléatoires, tel que :*

$$\mathbf{Y} \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2\end{pmatrix}, \begin{pmatrix}\Sigma_{1,1} & \Sigma_{1,2} \\ \Sigma_{1,2}^\top & \Sigma_{2,2}\end{pmatrix}\right)$$

On suppose que $\Sigma_{1,1}$ est inversible. Alors, la distribution de $\mathbf{Y}_2$ sachant $\mathbf{Y}_1$ est un vecteur gaussien où :

$$\mathbb{E}[\mathbf{Y}_2|\mathbf{Y}_1 = \mathbf{y}_1] = \boldsymbol{\mu}_2 + \Sigma_{1,2}^\top \Sigma_{1,1}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1) \tag{2.2}$$

$$\mathbb{C}\text{ov}[\mathbf{Y}_2|\mathbf{Y}_1 = \mathbf{y}_1] = \Sigma_{2,2} - \Sigma_{1,2}^\top \Sigma_{1,1}^{-1}\Sigma_{1,2} \tag{2.3}$$

2.1.2 Les processus gaussiens

Les processus stochastiques ou processus aléatoires peuvent être vus comme une généralisation des vecteurs aléatoires ou des variables aléatoires multivariées. Il existe plusieurs types de processus aléatoires dont les processus gaussiens, qui sont vus comme une généralisation des vecteurs gaussiens.

Définition 2.3 (Processus gaussien)**.** *Soient $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et*

$$\begin{array}{rccc} Y & : & \Omega \times \mathcal{X} & \to & \mathbb{R} \\ & & (\omega, \mathbf{x}) & \mapsto & Y(\omega, \mathbf{x}) \end{array},$$

avec $\mathcal{X} \subseteq \mathbb{R}^d$, $d \geq 1$. On dit que Y est un processus gaussien sur $\mathcal{X}$ lorsque :

$\forall n \in \mathbb{N}$, $\forall \mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{X}$, la fonction $\omega \mapsto (Y(\omega, \mathbf{x}_1), \ldots, Y(\omega, \mathbf{x}_n))$ est un vecteur gaussien.

Remarque 2.3. *$\forall \omega \in \Omega$, $\mathbf{x} \mapsto Y(\omega, \mathbf{x})$ est une fonction de $\mathcal{X} \to \mathbb{R}$. Un processus gaussien est donc une fonction aléatoire. On appelle $\mathbf{x} \mapsto Y(\omega, \mathbf{x})$ une trajectoire (ou une réalisation) de Y.*

Pour revenir au problème de la fonction $f : \mathcal{X} \subseteq \mathbb{R}^d \to \mathbb{R}$, on suppose que $f(\mathbf{x}) = Y(\omega^*, \mathbf{x})$, $\forall \mathbf{x} \in \mathcal{X}$ pour un événement $\omega^* \in \Omega$. La fonction f est donc une réalisation de la fonction aléatoire Y. Dans la suite, on écrira souvent $Y(\mathbf{x})$ au lieu de $Y(\omega, \mathbf{x})$.

Définition 2.4 (fonction moyenne)**.** *Soit Y un processus gaussien (PG) sur $\mathcal{X} \subseteq \mathbb{R}^d$.*
La fonction :

$$\begin{array}{rccc} \mu & : & \mathcal{X} & \to & \mathbb{R} \\ & & \mathbf{x} & \mapsto & \mathbb{E}[Y(\mathbf{x})] \end{array}$$

est la fonction moyenne de Y.

Définition 2.5 (fonction de covariance ou noyau)**.** *Soit Y un processus gaussien (PG) sur $\mathcal{X} \subseteq \mathbb{R}^d$.*
La fonction :

$$\begin{array}{rccc} k & : & \mathcal{X}\times\mathcal{X} & \to & \mathbb{R} \\ & & (\mathbf{x},\,\mathbf{y}) & \mapsto & \mathbb{C}\mathrm{ov}(Y(\mathbf{x}),Y(\mathbf{y})) \end{array}$$

est la fonction de covariance de Y.

Remarque 2.4. *Les notions de matrice de variance-covariance symétrique et semi-définie positive sont étendues pour les fonctions de covariance. Ainsi, on a que :*

- *$k(.,.)$ est symétrique si : $\forall \mathbf{x},\mathbf{y}\in\mathcal{X},\ k(\mathbf{x},\mathbf{y})=k(\mathbf{y},\mathbf{x})$.*
- *$k(.,.)$ est semi-définie positive si :*
  $\forall n\in\mathbb{N},\ \forall(\mathbf{x}_1,\ldots,\mathbf{x}_n)\in\mathcal{X}^n,\ \forall\boldsymbol{\alpha}\in\mathbb{R}^n,\ \sum_{i=1}^n\sum_{j=1}^n\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j)\geq 0$,
  i.e. la matrice $(k(\mathbf{x}_i,\mathbf{x}_j))_{1\leq i,j\leq n}$ est semi-définie positive.

Définition 2.6 (Stationnarité d'ordre 2)**.** *Une fonction aléatoire Y sur $\mathcal{X}$ est dite stationnaire d'ordre 2 si*

- *sa fonction moyenne est constante : $\forall(x,y)\in\mathcal{X}^2,\ \mu(x)=\mu(y)$.*
- *sa fonction de covariance est invariante par translation : $\forall\delta\in\mathcal{X},\ k(\mathbf{x},\mathbf{x}+\delta)=\kappa(\delta)$, où κ est une fonction $\mathcal{X}\to\mathbb{R}$.*

Un processus gaussien étant entièrement défini par sa moyenne et sa fonction de covariance, alors la stationnarité d'ordre 2 est équivalente à la stationnarité d'un processus gaussien. C'est-à-dire que la loi de distribution de Y est invariante par translation :

$\forall n\in\mathbb{N},\ \forall(\mathbf{x}_1,\ldots,\mathbf{x}_n)\in\mathcal{X}^n,\ \forall\delta\in\mathcal{X}$ tel que $\mathbf{x}_1+\delta,\ldots,\mathbf{x}_n+\delta\in\mathcal{X}$, alors la loi de $(Y(\mathbf{x}_1),\ldots,Y(\mathbf{x}_n))^\top$ est la même que la loi de $(Y(\mathbf{x}_1+\delta),\ldots,Y(\mathbf{x}_n+\delta))^\top$.

2.1.3 Le krigeage ou la régression par processus gaussien

Le krigeage (ou la régression par processus gaussien (RPG)) est très utilisé en géostatistique pour interpoler des données spatiales (see [Krige, 1951]). Les détails mathématiques peuvent être trouvés dans [Rasmussen and Williams, 2006]. On rappelle que l'objectif est de prédire les sorties d'une fonction $f:\mathcal{X}\to\mathbb{R}$ en se basant sur n observations : $(\mathbf{x}_1,f(\mathbf{x}_1)),\ldots,(\mathbf{x}_n,f(\mathbf{x}_n))$. On assimile la sortie de f à une réalisation d'un processus gaussien Y sur $\mathcal{X}$. Dans ce contexte, la loi de $Y_*=Y(\mathbf{x}^*)$, avec $\mathbf{x}^*\in\mathcal{X}$, est donnée par la loi conditionnelle aux observations $\mathbf{y}_n=(y_1,\ldots,y_n)^\top$, où $y_i=f(\mathbf{x}_i)$ pour $1\leq i\leq n$. $\mathbf{y}_n$ est vu comme une réalisation d'un vecteur gaussien $\mathbf{Y}_n$. Le vecteur $(\mathbf{Y}_n,Y_*)$ est aussi gaussien :

$$\begin{bmatrix}\mathbf{Y}_n\\ Y_*\end{bmatrix}=\mathcal{N}\left(\begin{bmatrix}\boldsymbol{\mu}_n\\ \mu(\mathbf{x}^*)\end{bmatrix},\begin{bmatrix}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n} & \mathbf{K}_{\mathbf{Y}_n,Y_*}\\ \mathbf{K}_{Y_*,\mathbf{Y}_n} & \mathbf{K}_{Y_*,Y_*}\end{bmatrix}\right)$$

avec $\boldsymbol{\mu}_n=(\mu(\mathbf{x}_1),\ldots,\mu(\mathbf{x}_n))^\top$ le vecteur moyenne de $\mathbf{Y}_n$, et $\mathbf{K}_{.,.}=\mathbb{C}\mathrm{ov}[.,.]$ qui satisfait $\mathbf{K}_{\mathbf{Y}_n,Y_*}=\mathbf{K}_{Y_*,\mathbf{Y}_n}^\top$. Finalement, d'après le TCG (Théorème 2.1), la loi de Y_* conditionnellement à $\mathbf{Y}_n=\mathbf{y}_n$ est donnée par :

$$Y_*|\{\mathbf{Y}_n = \mathbf{y}_n\} \sim \mathcal{N}(\hat{y}(\mathbf{x}^*), \sigma_y^2(\mathbf{x}^*)) \tag{2.4}$$

avec :

$$\begin{cases} \hat{y}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \mathbf{K}_{Y_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}(\mathbf{y}_n - \mu_n), \\ \sigma_y^2(\mathbf{x}^*) = \mathbf{K}_{Y_*,Y_*} - \mathbf{K}_{Y_*,\mathbf{Y}_n}\mathbf{K}_{\mathbf{Y}_n,\mathbf{Y}_n}^{-1}\mathbf{K}_{Y_*,\mathbf{Y}_n}^{\top} \end{cases} \tag{2.5}$$

$\hat{y}(\mathbf{x}^*)$ est un estimateur de $y(\mathbf{x}^*) = f(\mathbf{x}^*)$ (on dit aussi que $\hat{y}(.)$ est un méta-modèle de f). Un indicateur d'erreur du méta-modèle est de calculer l'intervalle de prévision de niveau α :

$$\text{IC}(\mathbf{x}^*) = [\hat{y}(\mathbf{x}^*) - q_{1-\alpha/2}\sigma_y(\mathbf{x}^*), \hat{y}(\mathbf{x}^*) + q_{1-\alpha/2}\sigma_y(\mathbf{x}^*)] \tag{2.6}$$

avec $q_{1-\alpha/2}$ est le quantile d'ordre $(1-\alpha/2)$ de la loi normale centrée-réduite. On a $q_{90\%} = 1.6449$, $q_{95\%} = 1.96$ et $q_{99\%} = 2.5758$.

2.1.4 Fonction moyenne et fonction de covariance

Pour utiliser le krigeage, il faut connaître la fonction moyenne de $\mathbf{Y}$, ainsi que sa fonction de covariance. La fonction moyenne permet de définir la tendance de l'estimation de la fonction objectif $f(\mathbf{x})$ (constante, linéaire, polynomiale, etc.). Dans la littérature, on peut trouver différents types de krigeage selon la fonction moyenne $\mu(\mathbf{x})$:

- le krigeage simple : quand $\mu(\mathbf{x}) = m$, avec $m \in \mathbb{R}$, où m est connu.
- le krigeage ordinaire : quand $\mu(\mathbf{x}) = m$, avec $m \in \mathbb{R}$, où m est inconnu.
- le krigeage universel : quand $\mu(\mathbf{x}) = \beta^{\top}\mathbf{h}(\mathbf{x})$ où $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \ldots, h_p(\mathbf{x}))$ est un vecteur de fonctions (h_i) connues, et β est un vecteur inconnu.

Dans le cas du krigeage ordinaire et universel, il faut alors estimer respectivement m et β. Les formules de moyenne conditionnelle et variance conditionnelle sont précisées dans [Cressie, 1993].

La fonction de covariance (ou noyau) est certainement l'ingrédient le plus important du krigeage. En effet, elle permet de contrôler la régularité de l'approximation de la fonction objectif $f(\mathbf{x})$. Comme dit dans la section 2.1.2, elle doit être symétrique et semi-définie positive. Une revue de fonctions de covariance communes est donnée dans [Rasmussen and Williams, 2006]. Quelques exemples de fonctions de covariance utilisées pour la régression par processus gaussien en 1D, sont donnés dans le Tableau 2.1.

La fonction de covariance est paramétrée par θ et σ^2. θ est appelé la longueur de corrélation. Intuitivement, il s'agit de la distance pour laquelle les observations de f sont fortement dépendantes. Ce paramètre d'échelle permet donc de contrôler les

Nom	Expression
Exponentielle	$k_{\sigma^2,\theta}(\mathbf{x},\mathbf{y}) = \sigma^2 \exp\left(-\frac{\lvert\mathbf{x}-\mathbf{y}\rvert}{\theta}\right)$
Matérn $\frac{5}{2}$	$k_{\sigma^2,\theta}(\mathbf{x},\mathbf{y}) = \sigma^2 \left(1+\frac{\sqrt{5}\lvert x-y\rvert}{\theta}+\frac{5(x-y)^2}{3\theta^2}\right)\exp\left(-\frac{\sqrt{5}\lvert x-y\rvert}{\theta}\right)$
Matérn $\frac{3}{2}$	$k_{\sigma^2,\theta}(\mathbf{x},\mathbf{y}) = \sigma^2 \left(1+\frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right)\exp\left(-\frac{\sqrt{3}\lvert x-y\rvert}{\theta}\right)$
Exponentielle au carré	$k_{\sigma^2,\theta}(\mathbf{x},\mathbf{y}) = \sigma^2 \exp\left(-\frac{(\mathbf{x}-\mathbf{y})^2}{2\theta^2}\right)$

Tableau 2.1 – Exemples de fonctions de covariance utilisées pour la régression par processus gaussien en 1D.

« fréquences d'oscillation » du processus gaussien. σ^2 est le paramètre de variance, et permet de contrôler l'amplitude du processus gaussien. θ et σ^2 doivent être estimés. Pour cela, on utilise l'estimation par maximum de vraisemblance.

Supposons par exemple que l'on connaît la fonction moyenne de $Y(\mathbf{x})$ et que cette fonction est nulle. Le processus centré dépend seulement de sa fonction de covariance qui dépend de σ^2 et θ. La vraisemblance est donnée par l'équation 2.7.

$$L(\sigma^2,\theta) = \frac{1}{(2\pi)^{n/2}|K_{\sigma^2,\theta}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{y}_n^\top K_{\sigma^2,\theta}^{-1}\mathbf{y}_n\right) \tag{2.7}$$

avec $K_{\sigma^2,\theta} = (k_{\sigma^2,\theta}(\mathbf{x}_i,\mathbf{x}_j))_{1\leq i,j\leq n}$. On estime θ et σ^2 respectivement par $\hat{\theta}$ et $\hat{\sigma}^2$ tel que :

$$(\hat{\sigma}^2,\hat{\theta}) \in \underset{(\sigma^2,\theta)\in\Theta}{\operatorname{argmax}}\ L(\sigma^2,\theta) \tag{2.8}$$

où Θ est un sous-ensemble de $\mathbb{R}^2$. En maximisant (2.7), on trouve l'ensemble des paramètres (σ^2,θ) qui améliore la capacité du modèle à expliquer les données. Pour simplifier le problème d'optimisation, il est commun d'utiliser le logarithme de la vraisemblance :

$$\log\left(L(\sigma^2,\theta)\right) = -\frac{1}{2}\log 2\pi - \frac{1}{2}\log\left(|K_{\sigma^2,\theta}|\right) - \frac{1}{2}\mathbf{y}_n^\top K_{\sigma^2,\theta}^{-1}\mathbf{y}_n \tag{2.9}$$

L'estimateur par maximum de vraisemblance est donc donné par :

$$(\hat{\sigma}^2,\hat{\theta}) \in \underset{(\sigma^2,\theta)\in\Theta}{\operatorname{argmin}}\ \log\left(|K_{\sigma^2,\theta}|\right) + \mathbf{y}_n^\top K_{\sigma^2,\theta}^{-1}\mathbf{y}_n \tag{2.10}$$

D'autres méthodes basées sur la validation croisée ([Hastie et al., 2009]) peuvent également être utilisées pour estimer (σ^2,θ) (voir [Bachoc, 2013] et [Rasmussen and Williams, 2006]). Dans les prochains chapitres, les paramètres (σ^2,θ) des modèles de krigeage, seront estimés par maximum de vraisemblance.

2.1.5 Modèle de krigeage avec entrées catégorielles

Dans cette section, nous suivons la présentation de [Roustant et al., 2020], section 2.

On se concentre sur la problématique des variables catégorielles en entrée du modèle. Un modèle de krigeage est défini par sa fonction de covariance (voir la section 2.1.4). Cependant, la question de la construction d'un noyau de covariance valide se pose lorsque des variables catégorielles sont en entrées du modèle [Zhang and Notz, 2015].

2.1.5.1 Processus gaussien avec des variables continues et catégorielles en entrée

On considère un ensemble de I variables continues $x_1, \ldots, x_I$, définies sur l'ensemble $\mathcal{X}$, avec $\mathcal{X} \subseteq \mathbb{R}^I$, et un ensemble de J variables catégorielles $u_1, \ldots, u_J$ avec $L_1, \ldots, L_J$ niveaux. Pour chaque $j = 1, \ldots, J$, les niveaux de u_j sont numérotés $1, 2, \ldots, L_j$. On note $\mathbf{x} = (x_1, \ldots, x_I)$, $\mathbf{u} = (u_1, \ldots, u_J)$, et $\mathbf{w} = (\mathbf{x}, \mathbf{u})$.

On considère des modèles de régression PG, définis sur l'espace produit :

$$\mathcal{D} = \mathcal{X} \times \prod_{j=1}^{J} \{1, \ldots, L_j\}.$$

On note k le noyau d'un processus gaussien $Z(\mathbf{w})$ défini sur $\mathcal{D}$:

$$k(\mathbf{w}, \mathbf{w}') = \mathbb{C}\mathrm{ov}(Z(\mathbf{w}), Z(\mathbf{w}'))$$

Les noyaux sur $\mathcal{D}$ sont obtenus en combinant les noyaux sur $\mathcal{X}$ et $\prod_{j=1}^{J} \{1, \ldots, L_j\}$. On note k_{cont} le noyau pour les variables continues, et k_{cat} celui pour les variables catégorielles. Des combinaisons standards valides sont :

- le produit : $k(\mathbf{w}, \mathbf{w}') = k_{cont}(\mathbf{x}, \mathbf{x}')k_{cat}(\mathbf{u}, \mathbf{u}')$,
- la somme : $k(\mathbf{w}, \mathbf{w}') = k_{cont}(\mathbf{x}, \mathbf{x}') + k_{cat}(\mathbf{u}, \mathbf{u}')$,
- ANOVA : $k(\mathbf{w}, \mathbf{w}') = (1 + k_{cont}(\mathbf{x}, \mathbf{x}'))(1 + k_{cat}(\mathbf{u}, \mathbf{u}'))$.

À leur tour, k_{cont} et k_{cat} peuvent être définis en appliquant ces mêmes opérations à des noyaux à une dimension. On note $*$ les opérations : produit, somme et ANOVA.

- $k_{cont}(\mathbf{x}, \mathbf{x}') = k^1_{cont}(x_1, x'_1) * \cdots * k^I_{cont}(x_I, x'_I)$, où $k^i_{cont}(x_i, x'_i)$ désigne le noyau pour la i^e variable continue $(i = 1, \ldots, I)$.
- $k_{cat}(\mathbf{u}, \mathbf{u}') = k^1_{cat}(u_1, u'_1) * \cdots * k^J_{cat}(u_J, u'_J)$, où $k^j_{cat}(u_j, u'_j)$ désigne le noyau pour la j^e variable catégorielle $(j = 1, \ldots, J)$.

Pour les variables continues, des exemples de fonctions de covariance valides sont donnés dans le tableau 2.1. La question qui se pose maintenant est : qu'est-ce qu'un noyau valide $k^j_{cat}(u_j, u'_j)$ pour une variable catégorielle u_j ?

2.1.5.2 Noyaux de covariance pour une variable catégorielle

On considère ici une seule variable catégorielle u avec les niveaux $1, \ldots, L$ ($J = 1$). Un noyau valide pour u est une matrice semi-définie positive de dimension $L \times L$, notée $\mathbf{T}$. Une variable catégorielle peut être de deux types : ordinale ou nominale. Une variable catégorielle avec les niveaux ordonnés correspond à une variable ordinale. Les niveaux d'une variable nominale sont nommés, par exemple par des noms de couleurs.

Noyaux pour variables ordinales : Dans ce cas, les niveaux peuvent être vus comme une discrétisation d'une variable continue. Un PG de Y sur $\{1, \ldots, L\}$ peut alors être obtenu à partir d'un PG sur l'intervalle $[0, 1]$, en utilisant une transformation F (fonction non décroissante) : $Y(u) = Z(F(u))$. Par conséquent, la matrice de covariance $\mathbf{T}$ peut être écrite :

$$\mathbf{T}_{u,u'} = k_Z(F(u), F(u')),\ u, u' = 1, \ldots, L. \tag{2.11}$$

k_Z est un noyau de covariance pour une variable continue. $\mathbf{T}_{u,u'}$ dépend donc de la distance entre u et u' déformée par F.

Noyaux pour variables nominales On considère ici le cas homoscédastique : la diagonale de $\mathbf{T}$ est constante. Ici, les termes de la matrice $\mathbf{T}$ sont calibrés afin de définir la variance sur les niveaux de u. Il y a plusieurs paramétrisations générales d'une matrice définie positive pour $\mathbf{T}$, basée sur les décompositions spectrales et de Cholesky.

- La décomposition spectrale de $\mathbf{T}$ est $\mathbf{T} = \mathbf{PDP}^\top$, avec $\mathbf{D}$ diagonale et $\mathbf{P}$ orthogonale. Des paramétrisations standards de $\mathbf{P}$ sont détaillées dans [Khuri and Good, 1989] et [Shepard et al., 2015].
- La décomposition de Cholesky est $\mathbf{T} = \mathbf{LL}^\top$, avec $\mathbf{L}$ une matrice triangulaire inférieure [Pinheiro and Bates, 1996].

La paramétrisation générale de $\mathbf{T}$ décrite nécessite $O(L^2)$ paramètres. Des approches parcimonieuses peuvent être utilisées en ajoutant des hypothèses supplémentaires au modèle. La matrice $\mathbf{T}$ peut être représentée par une matrice isotrope (symétrie composée) [Pinheiro and Bates, 2009] :

$$\mathbf{T}_{u,u'} = \begin{cases} v, \text{ si } u = u' \\ c, \text{ si } u \neq u' \end{cases},\ c/v \in (-1/(L-1), 1)$$

avec v la variance et c la covariance. Toutes les paires de niveaux sont traitées également, ce qui est limitant, particulièrement quand $L \gg 1$. Plus de flexibilité est possible en considérant des groupes de niveaux, mais cela ne sera pas utile ici car les variables considérées dans la thèse sont binaires.

2.2 Analyse de sensibilité

Comme dans la section 2.1, on s'intéresse à un simulateur $f : \mathcal{X} \subseteq \mathbb{R}^d \to \mathbb{R}$. L'analyse de sensibilité permet d'identifier les entrées de f qui ont une forte influence sur sa valeur de sortie, et, inversement les entrées qui ont une influence moindre. Plus généralement, elle permet de hiérarchiser les d entrées selon l'importance qu'elles ont sur la sortie. Cela a pour objectif de connaître de manière plus précise les entrées importantes à connaître et celles qui peuvent être négligées.

Pour l'analyse de sensibilité, deux approches sont possibles : locale ou globale. L'analyse de sensibilité locale consiste à étudier la variabilité de $\mathbf{y} = f(\mathbf{x})$ selon une petite variation de $\mathbf{x}$ autour d'une valeur de référence $\mathbf{x}_0$. Dans la thèse, on s'intéresse à l'analyse de sensibilité globale. L'objectif est aussi d'étudier la variabilité de $f(\mathbf{x})$ induite par la variation de $\mathbf{x}$, mais sur l'ensemble de son domaine de variation. Il existe de nombreuses méthodes d'analyse de sensibilité globale ([Iooss, 2011], [Faivre et al., 2016]). Ici, on considère une des méthodes les plus utilisées basée sur la décomposition de la variance : les indices de Sobol ([Sobol, 1993]).

2.2.1 Décomposition ANOVA

On considère un vecteur $\mathbf{X} = (X_1, \dots, X_d)$ de variables aléatoires réelles indépendantes, avec pour lois de probabilité $\mathbb{P}_1, \dots, \mathbb{P}_d$. En pratique, ces lois peuvent modéliser une incertitude ou un avis sur les valeurs que prennent les X_l. On suppose que l'on a :

$$Y = f(\mathbf{X}) \tag{2.12}$$

avec f une fonction de carré intégrable. Alors, il existe une unique décomposition de f de la forme :

$$f(\mathbf{X}) = f_0 + \sum_{l=1}^{d} f_l(X_l) + \sum_{l_1<l_2} f_{l_1,l_2}(X_{l_1}, X_{l_2}) + \cdots + f_{1,\dots,d}(X_1, \dots, X_d) = \sum_{\omega \in S} f_\omega(\mathbf{X}_\omega) \tag{2.13}$$

telle que :

$$\int f_\omega(\mathbf{x}_\omega) d\mathbb{P}(x_l) = 0, \ \forall l \in \omega. \tag{2.14}$$

où $S = \mathcal{P}(\{1, \dots, d\})$ est l'ensemble des sous-ensembles de $\{1, \dots, d\}$, et $\mathbf{X}_\omega = (\mathbf{X}_l)_{(l \in \omega)}$ est le vecteur des variables d'entrées dont les indices sont dans $\omega \in S$. La condition (2.14) implique l'unicité de la décomposition (2.13). En notant $d\mathbb{P}_{-\omega} = \prod_{l, l \notin \omega} d\mathbb{P}_l(x_l)$, les termes de la décomposition sont donnés explicitement par récursion :

— pour $\omega = \emptyset$, $f_\omega(\mathbf{x}_\omega)$ est la constante $f_0 = \mathbb{E}[Y] = \int f(\mathbf{x}) d\mathbb{P}(\mathbf{x})$.

— pour $\omega = \{l\}$, avec $l \in \{1, \dots, d\}$,

$$f_\omega(\mathbf{x}_\omega) = f_l(x_l) = \mathbb{E}[Y | X_l = x_l] - f_0 = \int f(\mathbf{x}) d\mathbb{P}_{-l}(\mathbf{x}) - f_0$$

.

— pour $\omega = \{l, k\}$, avec $l, k \in \{1, \dots, d\}$, $l \neq k$,

$$\begin{aligned} f_\omega(\mathbf{x}_\omega) = f_{l,k}(x_l, x_k) &= \mathbb{E}[Y|X_l = x_l,\ X_k = x_k] - f_l(x_l) - f_k(x_k) - f_0 \\ &= \int f(\mathbf{x}) d\mathbb{P}_{-\{l,k\}}(\mathbf{x}) - f_l(x_l) - f_k(x_k) - f_0 \end{aligned}$$

— plus généralement, pour tout $\omega \in S$:

$$\begin{aligned} f_\omega(\mathbf{x}_\omega) &= \mathbb{E}[Y|\mathbf{X}_\omega = \mathbf{x}_\omega] - \textstyle\sum_{\omega' \subsetneq \omega} f_{\omega'}(\mathbf{x}_{\omega'}) \\ &= \int f(\mathbf{x}) d\mathbb{P}_{-\omega}(\mathbf{x}) - \textstyle\sum_{\omega' \subsetneq \omega} f_{\omega'}(\mathbf{x}_{\omega'}) \end{aligned}$$

Les variables aléatoires X_l étant aléatoires et mutuellement indépendantes, (2.13) permet d'obtenir la décomposition de la variance fonctionnelle :

$$\mathbb{V}\text{ar}[Y] = \mathbb{V}\text{ar}\left[\sum_{\omega \in S} f_\omega(\mathbf{X}_\omega)\right] = \sum_{\omega \in S} \mathbb{V}\text{ar}[f_\omega(\mathbf{X}_\omega)] \tag{2.15}$$

L'équation (2.15) est appelée décomposition ANOVA (**AN**alysis **O**f **VA**riance).

2.2.2 Indices de sensibilité de Sobol

À partir de (2.15), des indices de sensibilité sont définis par :

$$\text{SI}_\omega = \frac{\mathbb{V}\text{ar}[f_\omega(\mathbf{X}_\omega)]}{\mathbb{V}\text{ar}[Y]} \tag{2.16}$$

On les appelle les indices de Sobol. L'indice de Sobol mesure la proportion de variance expliquée par un groupe de variables $\{X_l, l \in \omega\}$, avec $\omega \in S$. Plus précisément :

— Si $\omega = \{l\}$ alors $\text{SI}_\omega = \text{SI}_l$ est l'indice de Sobol associé à l'effet principal de la variable X_l sur Y.
— Si $\omega = \{l_1, \dots, l_k\}$ avec $k > 1$, SI_ω est l'indice de Sobol associé aux effets directs et aux interactions des variables $(X_{l_1}, \dots, X_{l_k})$ sur Y.

Ces indices sont compris entre 0 et 1 et leur somme vaut 1. On a une décomposition de la variance de la réponse ($\mathbb{V}\text{ar}[Y]$) en fractions correspondant aux effets principaux ou interactions des variables d'entrée. Pour un simulateur de d entrées, le nombre d'indices de sensibilité est $2^d - 1$. Lorsque d augmente, l'estimation et l'interprétation de tous ces indices deviennent vite impossible. Homma et Saltelli [Homma and Saltelli, 1996] ont donc introduit les indices de sensibilité totaux pour exprimer tous les effets d'une variable d'entrée sur la sortie.

$$\text{TSI}_l = \sum_{k \in S_l} \text{SI}_k \tag{2.17}$$

où S_l représente les sous-ensembles d'indices contenant l'indice l. (2.17) est la somme de tous les indices de Sobol faisant intervenir la variable X_l.

2.2.3 Estimation des indices de sensibilité

En pratique, les indices de Sobol doivent être estimés à partir d'un nombre fini d'évaluations de f. Pour cela, la méthode « pick−freeze » basée sur des échantillons Monte Carlo a été développée ([Sobol, 1993], [Saltelli, 2002]). Soient $\mathbf{x} = (\mathbf{x}_\omega, \mathbf{x}_{-\omega})$ et $\mathbf{x}' = (\mathbf{x}_\omega, \mathbf{x}'_{-\omega})$ deux échantillons indépendants de taille N, avec $\mathbf{x}_\omega$ l'échantillon des valeurs prises par $\mathbf{X}_\omega$, et $\mathbf{x}_{-\omega}$ celui des autres entrées. Ces échantillons ont été générés selon les lois de probabilité $(\mathbb{P}_\omega, \mathbb{P}_{-\omega})$, avec $\mathbb{P}_\omega$ les lois de probabilité associées au variables aléatoires $\mathbf{X}_\omega$, et $\mathbb{P}_{-\omega}$ celles des autres variables d'entrées. $\mathbb{V}\mathrm{ar}[f_\omega(\mathbf{X}_\omega)]$ défini dans (2.16) est calculé selon la formule suivante :

$$\mathbb{V}\mathrm{ar}[f_\omega(\mathbf{X}_\omega)] = \int f(\mathbf{x}) f(\mathbf{x}_\omega, \mathbf{x}'_{-\omega}) d\mathbf{x} d\mathbf{x}'_{-\omega} - f_0^2 \tag{2.18}$$

avec $f_0 = \int f(\mathbf{x}) d\mathbf{x}$. L'estimateur par Monte Carlo de $\mathbb{V}\mathrm{ar}[f_\omega(\mathbf{X}_\omega)]$ est donc :

$$\mathbb{V}\mathrm{ar}[f_\omega(\mathbf{X}_\omega)] \approx \frac{1}{N} \left[\sum_{k=1}^{N} f(\mathbf{x}_{\omega,(k)}, \mathbf{x}_{-\omega,(k)}) f(\mathbf{x}_{\omega,(k)}, \mathbf{x}'_{-\omega,(k)}) \right] - (\hat{f}_0^2) \tag{2.19}$$

avec $\hat{f}_0^2 = \frac{1}{2N} \left[\sum_{k=1}^{N} \left(f(\mathbf{x}_{\omega,(k)}, \mathbf{x}_{-\omega,(k)}) + f(\mathbf{x}_{\omega,(k)}, \mathbf{x}'_{-\omega,(k)}) \right) \right]$ et k l'indice de l'observation de l'échantillon. L'estimateur par Monte Carlo de (2.16) est donc :

$$\mathrm{SI}_\omega \approx \frac{\frac{1}{N} \left[\sum_{k=1}^{N} f(\mathbf{x}_{\omega,(k)}, \mathbf{x}_{-\omega,(k)}) f(\mathbf{x}_{\omega,(k)}, \mathbf{x}'_{-\omega,(k)}) \right] - (\hat{f}_0^2)}{\hat{\sigma}^2} \tag{2.20}$$

où $\hat{\sigma}^2 = \frac{1}{2N} \left[\sum_{k=1}^{N} \left(f(\mathbf{x}_{\omega,(k)}, \mathbf{x}_{-\omega,(k)})^2 + f(\mathbf{x}_{\omega,(k)}, \mathbf{x}'_{-\omega,(k)})^2 \right) \right] - (\hat{f}_0^2)$ est l'estimateur par Monte Carlo de $\mathbb{V}\mathrm{ar}[Y]$.

Cependant, ces méthodes sont très coûteuses en nombre de simulations du modèle f. D'autres méthodes ont alors été développées pour réduire le coût de l'estimation :

- L'utilisation d'échantillons de type quasi-Monte Carlo [Saltelli et al., 2008] (par exemples les suites de Sobol).
- La méthode FAST [Cukier et al., 1978], basée sur une transformée de Fourier multi-dimensionnelle de $f(.)$. Elle a été étendue au calcul d'indice total par [Saltelli et al., 2008]. Cependant, cette méthode ne supporte pas la montée en dimension des entrées [Tissot and Prieur, 2012].
- Dans le cas où f est coûteux en temps de calcul, une solution est de remplacer le simulateur par un méta-modèle, qui va prédire la sortie du modèle en un temps négligeable. Par exemple, les indices de Sobol peuvent être calculés en utilisant les processus gaussiens [Marrel et al., 2009].
- Une méthode récemment développée propose une approche basée sur les statistiques de rangs [Gamboa et al., 2020].

2.3 Analyse en composante principale fonctionnelle (ACPF)

L'Analyse en composante principale fonctionnelle (ACPF) est généralement utilisée en analyse de données fonctionnelles (ADF) [Ramsay and Silverman, 2004] afin de trouver les modes de variation dominantes dans un ensemble de fonctions, ici des cartes à deux dimensions. Ces modes correspondent à une base de dimension finie plus petite, où les données peuvent être représentées. Par analogie avec l'ACP, qui diagonalise la matrice de covariance numérique des données, l'ACPF diagonalise un opérateur de covariance.

Plusieurs approches sont possibles pour implémenter l'ACPF. Une méthode simple est de discrétiser les fonctions en m valeurs $\mathbf{z}_j$ également espacées dans $\mathcal{Z}$. Cela donnerait une matrice de données $\mathbf{Y}$ de dimension $n \times m$ sur laquelle une ACP multivariée standard peut être appliquée. Cependant, l'ACP traite les $(\mathbf{Y}_{i,j} = y_i(\mathbf{z}_j))_{j=1,\dots,m}$ indépendamment, et ne tient pas en compte de la dépendance spatiale. Dans la littérature, deux principales approches ont été proposées pour appliquer l'ACPF tout en prenant en compte le caractère fonctionnel des données.

Soit $Y(\mathbf{z})$, $\mathbf{z} \in \mathcal{Z}$, une fonction de carré intégrable aléatoire de moyenne nulle et de covariance $K(\mathbf{z}, \mathbf{z}') = \mathbb{C}\mathrm{ov}\left[Y(\mathbf{z}), Y(\mathbf{z}')\right]$. $\mathcal{Z}$ peut être un intervalle de temps ou un domaine spatial, par exemple. Ici, $\mathcal{Z}$ est un domaine spatial. On suppose observer $n \in \mathbb{N}$ réalisations de $Y(\mathbf{z})$: $\{y_1(\mathbf{z}), \dots, y_n(\mathbf{z})\}$.

2.3.1 Définition de la base des fonctions propres

Dans le cas des données multivariées, le concept de l'ACP est basé sur la combinaison linéaire des variables. L'ACP consiste à trouver une base orthonormée telle que la variance des projections est maximisée. On montre que cela revient à diagonaliser la matrice de covariance empirique. Plus de détails sur l'ACP se trouvent dans [Jolliffe, 2002]. Dans le cas des données fonctionnelles, la matrice est un opérateur, et les vecteurs propres des fonctions propres. L'approche de l'ACPF est similaire à celle de l'ACP. Mais étant dans un espace fonctionnel, on remplace la somme par une intégrale pour définir le produit scalaire (2.21).

$$t_{i,l} =< \xi_l, y_i >= \int \xi_l(\mathbf{z}) y_i(\mathbf{z}) d\mathbf{z}, \text{ avec } i = 1, \dots, n \tag{2.21}$$

où $t_{i,l}$ est la coordonnée (aussi appelée score) de $y_i(\mathbf{z})$ sur la composante (fonction propre) $\xi_l(\mathbf{z})$. On cherche donc $(\xi_l(\mathbf{z}))_l$ tel que :

La première composante $\mathbf{z} \mapsto \xi_1(\mathbf{z})$ maximise la variance des projections $(t_{i,1})_{1,\leq i \leq n}$:

$$\xi_1 \mapsto \frac{1}{n} \sum_{i=1}^{n} \left(\int \xi_1(\mathbf{z}) y_i(\mathbf{z}) d\mathbf{z} \right)^2 \tag{2.22}$$

sous la contrainte

$$||\xi_1||^2 = \int \xi_1(\mathbf{z})^2 d\mathbf{z} = 1 \tag{2.23}$$

La $l-$ème composante, $\mathbf{z} \mapsto \xi_l(\mathbf{z})$ $l \geq 2$, maximise la variance des $(t_{i,l})_{1,\leq i \leq n}$:

$$\xi_l \mapsto \frac{1}{n} \sum_{i=1}^{n} \left(\int \xi_l(\mathbf{z}) y_i(\mathbf{z}) d\mathbf{z} \right)^2 \tag{2.24}$$

sous les contraintes

$$||\xi_l||^2 = 1 \text{ et } \int \xi_l(\mathbf{z}) \xi_{l'}(\mathbf{z}) = < \xi_l, \xi_{l'} > = 0, \ l' < l \tag{2.25}$$

La première étape qui consiste à maximiser la variance, permet d'identifier le mode de variation le plus dominant dans l'ensemble de fonctions $\{y_1(\mathbf{z}), \ldots, y_n(\mathbf{z})\}$. La condition d'orthogonalité aide à identifier des modes de variation différentes. L'importance de la variance expliquée décroît à chaque étape (composante).

Résoudre ce problème d'optimisation multiple est équivalent à trouver les fonctions propres de la fonction de covariance $K(\mathbf{z}, \mathbf{z}')$ [Ramsay and Silverman, 2004], i.e. à résoudre :

$$\int K(\mathbf{z}, \mathbf{z}') \xi(\mathbf{z}') d\mathbf{z}' = \lambda \xi(\mathbf{z}) \tag{2.26}$$

Lorsqu'elles existent, les solutions de (2.26) sont notées $\lambda_1 \geq \lambda_2 \geq ...$, les valeurs propres, et (ξ_l), les fonctions propres correspondantes. Comme $(\xi_l)_l$ est une base orthonormée, la fonction aléatoire $Y(\mathbf{z})$ peut alors être décomposée comme suit :

$$Y(\mathbf{z}) = \sum_{l=1}^{+\infty} t_l \xi_l(\mathbf{z}) \tag{2.27}$$

où les $t_l = \int \xi_l(\mathbf{z}) Y(\mathbf{z}) d\mathbf{z}$, $l \geq 1$, sont des variables aléatoires décorrélées de moyenne nulle et de variance λ_l. Elles correspondent aux scores de $Y(\mathbf{z})$ dans la base de fonctions propres $(\xi_l)_{l \geq 1}$.

2.3.2 Calcul de l'ACPF

Pour pouvoir calculer l'ACPF, on se ramène à un espace fonctionnel de dimension finie.

Une première approche consiste à utiliser un noyau de covariance prédéfini k dans un espace de fonction $\mathcal{H}$, appelé RKHS [Rasmussen and Williams, 2006]. L'ACPF correspond à une troncature de la décomposition Karhunen-Loève de k (voir [Lamboni et al., 2011]),

obtenue en gardant les L premiers termes. En notant λ_l les valeurs propres et ξ_l les fonctions propres associées, on a alors :

$$k(\mathbf{x}, \mathbf{x}') = \sum_{l=1}^{L} \lambda_l \xi_l(\mathbf{x}) \xi_l(\mathbf{x}') \tag{2.28}$$

Ici, nous avons choisi d'utiliser une autre approche, proposée dans [Ramsay and Silverman, 2004]. Par analogie avec la matrice de covariance, $K(\mathbf{z}, \mathbf{z}')$ est défini comme suit :

$$K(\mathbf{z}, \mathbf{z}') = \frac{1}{n} \sum_{i=1}^{N} y_i(\mathbf{z}) y_i(\mathbf{z}') \tag{2.29}$$

Pour réduire l'équation propre (2.26) sous une forme discrétisée ou matricielle, chaque y_i est décomposé dans une base de fonctions $\boldsymbol{\phi}(\mathbf{z}) = (\phi_1(\mathbf{z}), \ldots, \phi_K(\mathbf{z}))^\top$.

$$y_i(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k^{(i)} \phi_k(\mathbf{z}) = {\boldsymbol{\alpha}^{(i)}}^\top \boldsymbol{\phi}(\mathbf{z}) \tag{2.30}$$

où $\boldsymbol{\alpha}^{(i)} = (\alpha_1^{(i)}, \ldots, \alpha_K^{(i)})^\top$ sont les coefficients de $y_i(\mathbf{z})$ dans la base de fonctions. On peut écrire cela de manière plus compacte en définissant le vecteur $\mathbf{Y}$ avec les composantes $y_1(\mathbf{z}), \ldots, y_n(\mathbf{z})$. La décomposition simultanée des n fonctions $y_i(\mathbf{z})$ peut alors être exprimée comme suit :

$$\mathbf{Y}(\mathbf{z}) = \mathbf{A}\boldsymbol{\phi}(\mathbf{z}) \tag{2.31}$$

où $\mathbf{A}$ est la matrice $n \times K$ des coefficients des décompositions sur la base $\boldsymbol{\phi}$: $\mathbf{A}_{i,k} = \alpha_k^{(i)}$. La fonction de covariance peut alors être exprimée de la façon suivante :

$$K(\mathbf{z}, \mathbf{z}') = n^{-1} \boldsymbol{\phi}(\mathbf{z})^\top \mathbf{A}^\top \mathbf{A} \boldsymbol{\phi}(\mathbf{z}') \tag{2.32}$$

Pour réduire l'équation propre (2.26), ξ est aussi décomposé dans la base $\boldsymbol{\phi}$.

$$\xi(\mathbf{z}) = \sum_{k=1}^{K} \beta_k \phi_k(\mathbf{z}) = \boldsymbol{\phi}(\mathbf{z})^\top \boldsymbol{\beta} \tag{2.33}$$

où $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)^\top$ sont les coefficients de $\xi(\mathbf{z})$ dans la base de fonctions. (2.32) et (2.33) donnent :

$$\int K(\mathbf{z}, \mathbf{z}') \xi(\mathbf{z}') d\mathbf{z}' = \boldsymbol{\phi}(\mathbf{z})^\top n^{-1} \mathbf{A}^\top \mathbf{A} \mathbf{G} \boldsymbol{\beta} \tag{2.34}$$

où $\mathbf{G} = \int \phi(\mathbf{z}) \phi(\mathbf{z})^\top d\mathbf{z}$ est la matrice de Gram de la base $\boldsymbol{\phi}(\mathbf{z})$. (2.26) peut alors être exprimée comme suit :

$$\boldsymbol{\phi}(\mathbf{z})^\top n^{-1} \mathbf{A}^\top \mathbf{A} \mathbf{G} \boldsymbol{\beta} = \lambda \boldsymbol{\phi}(\mathbf{z})^\top \boldsymbol{\beta} \tag{2.35}$$

(2.35) est vraie quel que soit $\mathbf{z}$. Ce qui implique que cette équation est équivalente à :

$$n^{-1}\mathbf{A}^\top\mathbf{AG}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \tag{2.36}$$

Afin d'obtenir un problème aux valeurs propres, on définit $\mathbf{u} = \mathbf{G}^{1/2}\boldsymbol{\beta}$ qui est donc solution de :

$$n^{-1}\mathbf{G}^{1/2}\mathbf{A}^\top\mathbf{AG}^{1/2}\mathbf{u} = \lambda\mathbf{u}. \tag{2.37}$$

L'ACP fonctionnelle est donc équivalent à diagonaliser la matrice $n^{-1}\mathbf{G}^{1/2}\mathbf{A}^\top\mathbf{AG}^{1/2}$. Notons que comme cette matrice est symétrique, elle est diagonalisable dans une base orthonormée. La diagonaliser revient à faire une ACP avec la métrique $\mathbf{G}$ comme produit scalaire dans $\mathbb{R}^K$: $< x, y >= x^\top\mathbf{G}y,\ x, y \in \mathbb{R}^K$. Si la base $\boldsymbol{\phi}$ est une base orthonormée, alors la matrice $\mathbf{G}$ est égale à la matrice identité $\mathbf{I}_K$. D'après (2.36), (2.26) est équivalent à :

$$n^{-1}\mathbf{A}^\top\mathbf{A}\boldsymbol{\beta} = \lambda\boldsymbol{\beta} \tag{2.38}$$

$n^{-1}\mathbf{A}^\top\mathbf{A}$ correspond à la matrice de covariance des coefficients de la base $\boldsymbol{\phi}(\mathbf{z})$. Résoudre (2.26) revient donc à la diagonaliser. On en déduit donc que l'ACPF est équivalent à appliquer une ACP standard multivariée aux coefficients $\mathbf{A}$. Ceci limite la dimension de la base de fonctions propres à K, dimension de la base $\boldsymbol{\phi}(\mathbf{z})$, à la place de la dimension infinie.

2.4 Bases de fonctions pour l'approximation de données spatiales

Le calcul de l'ACPF a besoin de déterminer une base de fonctions, où les données fonctionnelles sont approximées. Dans la thèse, on s'intéresse plus particulièrement à l'approximation de données spatiales, qui peuvent contenir des comportements locaux spécifiques comme pour les cartes de submersion marine : des irrégularités marquées dans les zones urbaines, expliquées par la présence d'infrastructures, des zones non inondées, etc. Des systèmes de bases existent pour représenter de telles données, avec une analyse de carte zone par zone. Parmi les méthodes ADF et de traitement d'images, les bases B-splines et d'ondelettes sont couramment utilisées.

2.4.1 La base B-splines

Les splines sont des fonctions par morceaux définies par des polynômes. Elles sont couramment utilisées pour approximer des données fonctionnelles non-périodiques. Des systèmes de base ont été développés pour les fonctions splines. Comme les cartes de submersion marine peuvent être irrégulières, on considère les bases B-splines de degré 1 [Ramsay and Silverman, 2004], qui définissent une base de fonctions linéaires par morceaux. Pour des données spatiales, des splines à deux dimensions sont obtenues par produit tensoriel. Des exemples sont montrés dans la Figure 2.1. Plus précisément, soient deux bases B-splines sur $[0, 1]$, notées $\phi^{(i)}(z_i) = (\phi_1^{(i)}(z_i), \ldots, \phi_{K_i}^{(i)}(z_i))^\top$, où i est le numéro

de la coordonnée ($i \in \{1, 2\}$), et K_i est le nombre de nœuds par coordonnée. On note $K = K_1 K_2$ le nombre de fonctions dans la base. Ensuite, des B-splines à deux dimensions sont obtenues par :

$$\phi_{k_1,k_2}(z_1, z_2) = \phi^{(1)}_{k_1}(z_1)\phi^{(2)}_{k_2}(z_2), \text{ with } 1 \leq k_i \leq K_i,\ i = 1, 2.$$

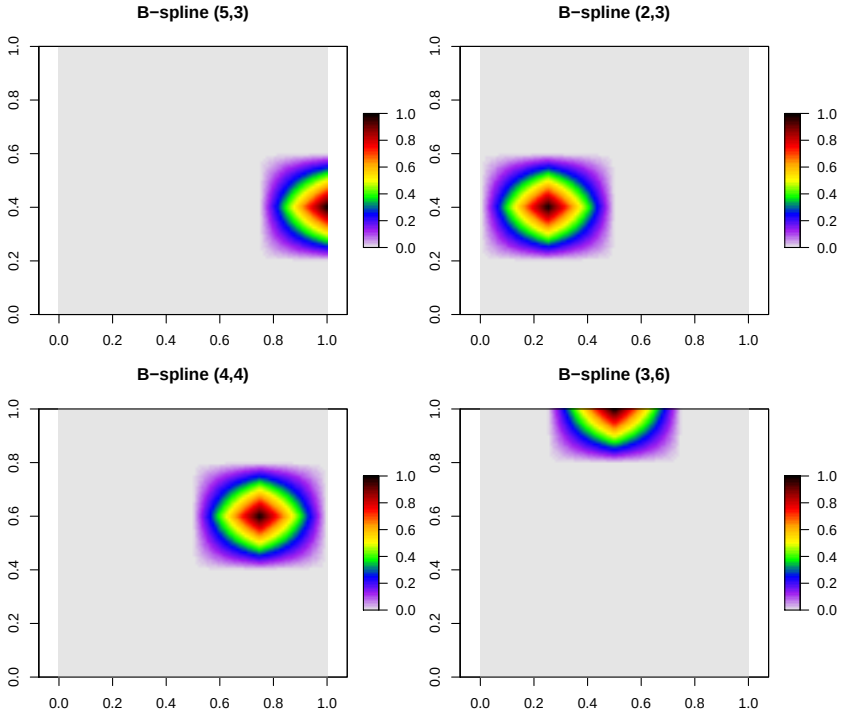

Figure 2.1 – Exemples de fonctions d'une base B-splines de degré 1, définie sur $[0, 1]^2$. La subdivision de l'axe x est $\left\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\right\}$. Celle de l'axe y est $\left\{0, \frac{1}{5}, \frac{2}{5}, \frac{3}{5}, \frac{4}{5}, 1\right\}$.

2.4.2 La base d'ondelettes

Les ondelettes sont des fonctions oscillantes définies sur un ensemble compact (i.e. l'oscillation existe dans une durée finie). Ce sont des fonctions de carré intégrable et moyenne nulle. Différents types d'ondelette existent, ce qui est un point clé de l'analyse par ondelette. Une base d'ondelettes est construite en utilisant des versions translatées et dilatées d'une ondelette « mère ». L'idée principale des ondelettes est d'analyser un signal (ou une image, ou une carte) selon des échelles multiples (ou résolutions) [Mallat, 1999, Mallat, 1989, Gençay et al., 2001]. Remarquons que pour l'analyse multi-résolution, on a besoin à une certaine échelle de compléter l'analyse fournie par les ondelettes, avec un ensemble de fonctions qui sont des versions translatées et dilatées d'une fonction « échelle », qui est associée à l'ondelette mère. À une échelle donnée, les coefficients associés à la fonction échelle sont calculés avec un filtre passe-bas, alors que ceux obtenus avec l'ondelette mère sont calculés avec un filtre passe-bande.

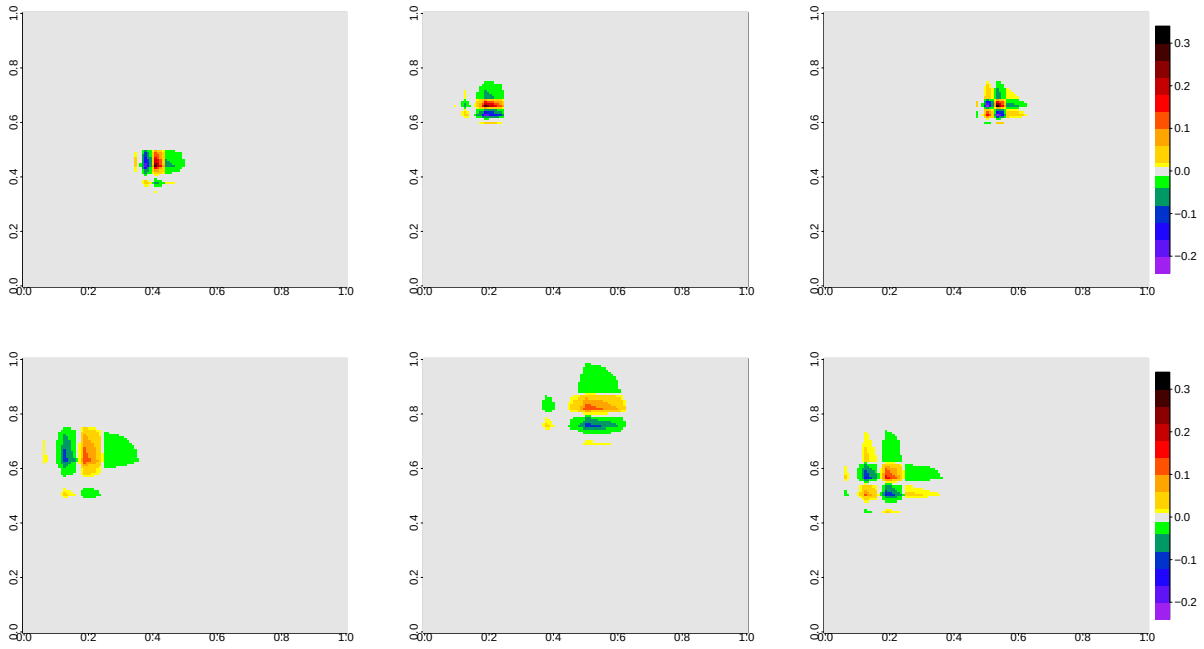

FIGURE 2.2 – Exemple d'ondelettes de Daubechies D4 sur $[0,1]^2$, qui sont discrétisées dans une grille de dimension 128×128. De gauche à droite, les figures représentent des exemples d'ondelettes horizontales, verticales et diagonales. Les figures du haut sont des ondelettes à l'échelle 3 (i.e. que la résolution de l'image est divisée par 3). Celles du bas sont des ondelettes à l'échelle 4.

Deuxième partie

Contribution en analyse de sensibilité pour les modèles avec sortie spatiale

Chapitre 3

Méta-modélisation de modèle avec sortie spatiale

Sommaire

3.1 Objectif et état de l'art

3.1.1 Formulation du problème

On considère le simulateur suivant :

$$\begin{array}{rccc} f & : & \mathcal{X} \subseteq \mathbb{R}^d & \to & \mathbb{L}^2(\mathcal{Z}) \\ & & \mathbf{x} & \mapsto & y_{\mathbf{x}}(\mathbf{z}) \end{array} \tag{3.1}$$

où $\mathbf{x}$ est le vecteur des entrées, et $\mathcal{Z}$ est un domaine spatial. $y_{\mathbf{x}}(\mathbf{z})$ est la sortie de f, une fonction de $\mathbb{L}^2(\mathcal{Z})$. Dans le cas des submersions marines, $y_{\mathbf{x}}(\mathbf{z})$ est assimilé à la valeur de la carte de sortie à la localisation $\mathbf{z}$. On suppose connaître n simulations de $f : \{(\mathbf{x}_i, y_{\mathbf{x}_i}(\mathbf{z})), i = 1, \ldots, n\}$. L'objectif du chapitre est de prédire $f(\mathbf{x}^\star)$ en un nouveau

point $\mathbf{x}^\star$, sans faire de nouveau appel au simulateur f.

Il existe divers méta-modèles, dans le cas où la sortie d'un simulateur est scalaire : régression linéaire, réseau de neurones, krigeage, etc. Cependant ici, la sortie est fonctionnelle. Ces méthodes ne sont donc pas directement applicables. Ici, on s'intéresse à étendre la méthode du krigeage au cas des sorties fonctionnelles. Nous nous focalisons sur cette méthode au vu de sa capacité à interpoler les données et à mesurer l'incertitude des prédictions aux points inconnus. De plus, cette méthode est malléable par sa fonction de covariance qui permet d'intégrer a priori des informations d'experts.

3.1.2 État de l'art

En pratique, $\mathcal{Z}$ est discrétisé en une grille $m_1 \times m_2$. Par exemple, dans la partie III, pour le cas des Bouchôleurs, en France, le modèle de submersion marine (appelé modèle ALEA) de la CCR estime des cartes de profondeurs d'eau, qui sont définies par une grille de taille 213×213, où chaque case correspond à une résolution 25m $\times$ 25m.

Après discrétisation du domaine $\mathcal{Z}$, une solution serait de considérer la sortie non pas sous une forme matricielle (grille de valeurs) mais sous une forme vectorielle. Ainsi, le problème se resitue dans le contexte d'un modèle avec une sortie multivariée. Pour ce type de modèle, des méthodes ont été développées comme celle du co-krigeage [Chiles and Delfiner, 2012, Wackernagel, 2013]. Le co-krigeage est une extension du krigeage (voir 2.1) au cas multivarié. Pour prédire la valeur d'un pixel de la carte, le co-krigeage va non seulement se baser sur sa variance, mais aussi sur la covariance croisée avec les autres pixels [López-Lopera et al., 2020]. Cependant, pour une bonne approximation de $\mathcal{Z}$, la discrétisation peut être de grande taille. L'inversion de matrice de covariance de grande dimension rend difficilement applicable le co-krigeage. Par exemple, les cartes de submersion des modèles ALEA du BRGM et de la CCR sont respectivement de dimension $194 \times 202 = 39\,188$ et $213 \times 213 = 45\,369$.

Une autre méthode résoud le problème de la sortie spatiale de la manière suivante (voir [Chen et al., 2011, Marrel et al., 2011]). En premier, la dimension de la sortie est réduite, le plus souvent par analyse en composante principale (ACP) [Jolliffe, 2002] (en considérant la sortie discrétisée sous forme vectorielle) ou en utilisant une décomposition sur une base de fonctions orthonormée (sur une base de Fourier ou d'ondelettes, par exemple). Cela permet d'obtenir un vecteur de sortie de plus petite dimension, soit formée des premières composantes (si on utilise l'ACP), soit des coefficients de la base de fonctions. En second, un méta-modèle est construit pour le nouveau vecteur de sortie. Habituellement, cela consiste à construire indépendamment un méta-modèle pour chaque coordonnée, considéré comme une sortie scalaire. Comme précisé dans la section 3.1.1, nous nous concentrons sur les modèles de régression par processus gaussien.

Cependant, ces méthodes sont difficilement applicables dans le contexte des modèles

ALEA, en raison de l'étape de réduction de dimension. En effet, comme mentionné au-dessus, le vecteur de la sortie spatiale peut être de longueur 39 188 ou 45 369. Cependant, l'ACP traite chaque variable spatiale indépendamment et ne tient pas compte de la dépendance spatiale. Par ailleurs, une carte de submersion marine présente de fortes variations locales, ce qui signifie que le niveau d'eau n'est pas une fonction lisse spatialement. Par conséquent, même en utilisant une base de fonctions adaptée telle que les ondelettes ou les B-splines, un grand nombre de coefficients doivent être modélisés pour avoir une approximation la plus précise possible. Cela réduit les avantages de la réduction de dimension.

3.1.3 Résumé de l'approche proposée

Pour aborder le problème, on propose d'utiliser l'ACPF (voir la section 2.3) couplée à une étape supplémentaire de réduction de dimension. L'ACPF est équivalent à réaliser une ACP sur les coefficients d'une décomposition sur une base de fonctions, avec la métrique donnée par la matrice de Gram de la base de fonctions. Elle peut être appliquée en utilisant des bases connues : Fourier, ondelettes, et B-splines, par exemple. Même s'il est possible d'appliquer l'ACPF en utilisant une base non-orthonormée, comme les B-splines, il est utile de l'orthonormaliser. En effet, afin de réduire encore plus la dimension, on sélectionne les éléments de la base de fonctions au moyen d'une décomposition de l'énergie des cartes (variance spatiale). Cependant, cette étape de sélection nécessite de travailler sur une base orthonormée. Sélectionner une partie des coefficients permet de réduire le nombre de variables pour l'ACP, afin de réduire le temps de calcul. Les coefficients non utilisés dans l'ACP sont estimés par moyenne empirique.

L'utilisation d'une base de fonctions et de l'ACP cumule leurs avantages. En effet, pour une base de fonctions adaptée, cela permet de considérer la dépendance spatiale des variables de la sortie du modèle. De plus, en combinant sélection des coefficients de la base de fonctions et ACP, la dimension est davantage réduite. Ainsi, on obtient un petit nombre de variables, et la construction d'un méta-modèle pour chacune d'elles est raisonnable en temps de calcul. L'indépendance des méta-modèles a du sens car les composantes principales sont décorrélées.

3.2 Procédure proposée

3.2.1 Description de l'algorithme

Dans cette section, une vue d'ensemble de la méthode proposée est donnée sous forme d'algorithme. Des détails méthodologiques des étapes clefs de la procédure seront donnés dans la suite du chapitre. Ce travail a fait l'objet d'un article dans la revue *Reliability Engineering and System Safety* pour « Special Issue on Sensitivity Analysis of Model Outputs » [Perrin et al., 2021].

3.2.1.1 Méthodologie avec une base orthonormée

On considère le problème défini dans la section 3.1. La sortie de f étant une fonction de $\mathbb{L}^2(\mathcal{Z})$, il est nécessaire de se ramener à la dimension finie. Au lieu de discrétiser le domaine spatial $\mathcal{Z}$ en utilisant un nombre fini de localisations, on considère un sous-espace fonctionnel de dimension fini. Cet espace est défini par une base de fonctions, que l'on note $\Phi(\mathbf{z}) = (\phi_1(\mathbf{z}), \dots, \phi_K(\mathbf{z}))^\top$, avec K, la taille de la base de fonctions. On suppose pour tout $\mathbf{x} \in \mathcal{X}$:

$$y_{\mathbf{x}}(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k(\mathbf{x})\phi_k(\mathbf{z}) = \boldsymbol{\alpha}(\mathbf{x})^\top \Phi(\mathbf{z}) \tag{3.2}$$

où $\boldsymbol{\alpha}(\mathbf{x}) = (\alpha_1(\mathbf{x}), \dots, \alpha_K(\mathbf{x}))^\top$ est le vecteur des coefficients. Prédire la carte $y_{\mathbf{x}^\star}(\mathbf{z})$ revient donc à prédire les K coefficients $\boldsymbol{\alpha}(\mathbf{x}^\star) = (\alpha_1(\mathbf{x}^\star), \dots, \alpha_K(\mathbf{x}^\star))^\top$.

K est choisi tel que la racine de la moyenne de l'erreur quadratique de l'approximation de $y_{\mathbf{x}}(\mathbf{z})$ dans la base $\Phi(\mathbf{z})$ soit minimisée. Cependant, afin d'approximer avec précision la sortie spatiale, la taille de la base de fonctions K peut *a priori* être grande. Pour réduire davantage la dimension, deux procédures sont appliquées successivement : la sélection des coefficients importants de la base et une ACP sur les coefficients sélectionnés.

On détaille l'étape de la sélection des coefficients qui nécessite un traitement minutieux. Dans un premier temps, on suppose que la base $\Phi(\mathbf{z})$ est une base orthonormée, mais une alternative basée sur une régression « sparse » est proposée dans la section 3.2.1.2. Lorsque la base n'est pas orthogonale, on peut l'orthonormaliser, en utilisant des méthodes comme Gram-Schmidt [Björck, 1994], ou une procédure spécifique pour les B-splines [Qin, 2000, Redd, 2012, Liu et al., 2019]. L'ACP fonctionnelle (ACPF) peut ensuite être appliquée en utilisant la nouvelle base. Ceci est en fait équivalent à l'ACPF en utilisant la base d'origine, comme indiqué dans la Propriété 3.1 ci-dessous.

Propriété 3.1 (ACPF et orthonormalisation). *L'ACPF pour une base* Φ *est équivalente à l'ACPF pour une base orthonormalisée obtenue à partir de* Φ*. En d'autres termes, appliquer l'ACPF aux coefficients de la base avec la métrique donnée par la matrice de Gram, notée* G*, équivaut à appliquer l'ACP aux coefficients de la base orthonormalisée avec la métrique donnée par la matrice identité. La matrice* G *est définie par :*

$$\mathrm{G} = \left(\int \phi_k(\mathbf{z}) \phi_{k'}(\mathbf{z}) d\mu(\mathbf{z}) \right)_{1 \leq k,k' \leq K}$$

Preuve Notons $\Phi = (\phi_1, \ldots, \phi_K)^\top$. Une base orthonormalisée obtenue à partir de Φ a la forme $\mathrm{R}^{-1}\Phi$, où R est une racine carrée de G, c'est-à-dire telle que $\mathrm{R}\mathrm{R}^\top = \mathrm{G}$ (voir par exemple, [Redd, 2012], Lemme 1). Ensuite, en utilisant l'isométrie $||c||^2_{\mathrm{G}} = c^\top \mathrm{G} c = ||\mathrm{R}^\top c||^2$, effectuer l'ACP avec la métrique G sur les coefficients de la base $c = (c_1, \ldots, c_K)^\top$ équivaut à effectuer l'ACP avec la métrique donnée par la matrice identité sur les coefficients transformés $\mathrm{R}^\top c$, qui sont les coefficients de la base orthonormée $\mathrm{R}^{-1}\Phi$. □

Pour faciliter l'écriture, la nouvelle base orthonormée est aussi notée $\Phi(\mathbf{z})$. Après l'orthonormalisation, on peut procéder à la sélection des coefficients. On remarque que l'énergie spatiale peut être décomposée comme suit :

$$||y_{\mathbf{x}}||^2_2 = \int y_{\mathbf{x}}(\mathbf{z})^2 d\mu(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k(\mathbf{x})^2 \tag{3.3}$$

Par conséquent, chaque coefficient $\alpha_k(\mathbf{x})$, $k = 1, \ldots, K$, correspond à une part de l'énergie. L'importance d'un coefficient peut alors être quantifiée par le ratio $\frac{\alpha_k(\mathbf{x})^2}{\sum_{k'=1}^{K} \alpha_{k'}(\mathbf{x})^2}$. Cependant, ce ratio dépend de $\mathbf{x}$, ce qui pose un problème pour la prédiction en un nouveau point $\mathbf{x}^\star$. Par conséquent, on considère à la place le ratio moyen :

$$\lambda_k = \mathbb{E}\left[\frac{\alpha_k(\mathbf{x})^2}{\sum_{k'=1}^{K} \alpha_{k'}(\mathbf{x})^2} \right] \tag{3.4}$$

λ_k est indépendant de $\mathbf{x}$. En pratique, on approxime l'espérance par la moyenne empirique sur l'ensemble d'apprentissage $(\mathbf{x}^{(i)})_{i=1,\ldots,n}$, qui est une bonne approximation si les points du plan d'expérience ont été tirés selon la loi μ dans Ω (par exemple, un design remplissant l'espace (en anglais : « space-filling design ») si μ est une distribution uniforme). Maintenant, les coefficients peuvent être ordonnés par ordre décroissant des $(\lambda_k)_{k=1,\ldots,K}$. On note (k), $k = 1, \ldots, K$, les indices des coefficients suivant ce nouveau classement. On tronque maintenant la base en sélectionnant les $\tilde{K}$ $(\leq K)$ premiers coefficients tel que $\sum_{k=1}^{\tilde{K}} \lambda_{(k)} \leq p$, où $p \in [0, 1]$ est un paramètre réglant la proportion moyenne de l'énergie. Les coefficients restants $\alpha_{(k)}(\mathbf{x}^\star)$, $k = (\tilde{K}+1), \ldots, K$ sont estimés par moyenne empirique : $\hat{\alpha}_{(k)}(\mathbf{x}^\star) = \frac{1}{n} \sum_{i=1}^{n} \alpha_{(k)}(\mathbf{x}^{(i)})$. On note que la complexité de cette étape de sélection est $O(nK)$, en incluant le calcul des λ_k et leur classement. Ceci est modéré et négligeable par rapport à la complexité de

la prochaine étape de l'ACP, en $O(\min(n, \tilde{K})^3)$.

Après l'étape de sélection, une ACP standard est appliquée sur les $\tilde{K}$ coefficients sélectionnés. On note n_{PC} le nombre de composantes principales retenues. Ensuite, chaque score des n_{PC} première composantes principales est estimé par des processus gaussiens indépendants, ce qui donnera les prédictions des coefficients $(\alpha_{(k)}(\mathbf{x}^\star))_{k=1,\dots,\tilde{K}}$. Les paramètres $\tilde{K}$ et n_{PC} sont choisis tels que l'erreur de prédiction du méta-modèle soit minimisée. Des méthodes comme la validation croisée [Hastie et al., 2009] peuvent être utilisées (par exemple, voir la section 3.3.2). On peut penser utiliser des processus gaussiens multivariés (comme les modèles de co-krigeage). Cependant, en plus d'augmenter la charge de calcul, leurs avantages peuvent être limités en comparaison à des processus gaussiens séparés car les composantes principales sont décorrélées.

L'ensemble de la méthodologie est résumé dans l'algorithme 1. En pratique, les paramètres p et n_{PC} sont ajustés par validation croisée. Au lieu de p, on peut aussi directement calibrer la taille de la troncature $\tilde{K}$.

Algorithme 1 Objectif : Prédire $f(\mathbf{x}^\star) = y_{\mathbf{x}^\star}(\mathbf{z})$, $\mathbf{z} \in \mathcal{Z}$

Entrées: $\{(\mathbf{x}_i, y_{\mathbf{x}_i}(\mathbf{z})), i = 1, \dots, n\}$; $\Phi(\mathbf{z}) = (\phi_1(\mathbf{z}), \dots, \phi_K(\mathbf{z}))^\top$; p (proportion de l'énergie moyenne) ; n_{PC} (nombre de composantes principales)

Sortie: $\hat{f}(\mathbf{x}^\star) = \hat{y}_{\mathbf{x}^\star}(\mathbf{z})$

1. Si $\Phi(\mathbf{z})$ n'est pas une base orthonormée, l'orthonormaliser en utilisant une méthode adaptée. Pour simplifier l'écriture, la nouvelle base est toujours noté $\Phi(\mathbf{z})$.
2. Décomposer les $(y_{\mathbf{x}^{(i)}}(\mathbf{z}))_{i=1,\dots,n}$ dans la base $\Phi(\mathbf{z})$.
3. Trier les coefficients $(\alpha_k(\mathbf{x}))_{k=1,\dots,K}$ selon l'ordre décroissant du critère (3.4). On note (k), $k = 1, \dots, K$, les indices des coefficients suivant le nouvel ordre. Puis, sélectionner les $\tilde{K} \ll K$ coefficients les plus importants tel que $\sum_{k=1}^{\tilde{K}} \lambda_{(k)} \leq p$
4. Appliquer l'ACP dans $\mathbb{R}^{\tilde{K}}$ sur le jeu de données des coefficients évalués aux points du plan d'expérience $(\alpha(\mathbf{x}^{(i)}))_{i=1,\dots,n}$. Les n_{PC} premières composantes principales sont choisies. Les coordonnées dans les premières composantes principales sont notées $t_1(\mathbf{x}_i), \dots, t_{n_{PC}}(\mathbf{x}_i)$, $i = 1, \dots, n$, et $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{n_{PC}}$ désignent les vecteurs propres associés.
5. Pour chaque composante principale $l = 1, \dots, n_{PC}$, prédire $t_l(\mathbf{x}^\star)$ (noté $\hat{t}_l(\mathbf{x}^\star)$) par régression GP (voir (2.4) et (2.5)), basée sur les observations $t_l(\mathbf{x}^{(i)})$ $(i = 1, \dots, n)$.
6. Prédire les coefficients $\alpha_k(\mathbf{x}^\star)$.
   — pour $k = \tilde{K} + 1, \dots, K$, $\hat{\alpha}_{(k)}(\mathbf{x}^\star) = \frac{1}{n} \sum_{i=1}^{n} \alpha_{(k)}(\mathbf{x}^{(i)})$.
   — pour $k = 1, \dots, \tilde{K}$, prédire les coefficients par leurs coordonnées prédites dans les composantes principales : $\hat{\alpha}_{(k)}(\mathbf{x}^\star) = \sum_{l=1}^{n_{PC}} \hat{t}_l(\mathbf{x}^\star)\boldsymbol{\omega}_l$.
7. Calculer la prédiction de $y_{\mathbf{x}^\star}(\mathbf{z})$ avec les coefficients prédits $\hat{\alpha}_k(x^\star)$ à partir de la formule de sa décomposition (3.2).

3.2.1.2 Variante, sans orthonormalisation

Les étapes 1 à 3 de l'algorithme 1 décrivent comment sélectionner les coefficients de la base avec une décomposition $\mathbb{L}^2$, qui a une interprétation physique en termes d'énergie. Alternativement, on peut penser à appliquer des techniques statistiques « sparse », basées sur une pénalité $\mathbb{L}^1$. Par exemple, la régression Lasso [Tibshirani, 1996] s'écrit comme suit (voir 3.5). Pour un $\mathbf{x} \in \mathcal{X}$ donné, soit $\lambda(\mathbf{x}) > 0$ un paramètre de pénalité (attention, il ne s'agit pas ici du critère basé sur l'énergie comme défini dans l'équation 3.4). Sans perte de généralité, nous supposons que $y_{\mathbf{x}}(\mathbf{z})$ a été centré par rapport à $\mathbf{z}$. Ensuite, les coefficients $\alpha_k(\mathbf{x})$ sont estimés en résolvant le problème de régression pénalisée :

$$\underset{\alpha_1(x),\dots,\alpha_K(x)}{min} ||y_{\mathbf{x}}(\mathbf{z}) - \sum_{k=1}^{K} \alpha_k(\mathbf{x})\phi_k(\mathbf{z})||^2 + \lambda(\mathbf{x}) \sum_{k=1}^{K} |\alpha_k(\mathbf{x})|. \tag{3.5}$$

En pratique la norme $\mathbb{L}^2$ est remplacée par sa discrétisation sur $\mathbf{z}$. Ce problème d'optimisation induit une parcimonie, et force certains coefficients $\alpha_k(\mathbf{x})$ à être égaux à zéro. Cependant, comme pour la sélection basée sur l'énergie, cette sélection dépend de $\mathbf{x}$. Par conséquent, un critère global doit être considéré, tel que $\mathbb{E}(\alpha_k(\mathbf{x}))$ ou $\mathbb{P}(\alpha_k(\mathbf{X}) \neq 0)$, pour faire le choix des valeurs importantes de k, indépendamment de $\mathbf{x}$. Ainsi, bien que la technique de Lasso induise une parcimonie pour une seule carte $y_{\mathbf{x}}(.)$, ce n'est pas vrai pour la collection de cartes (quand $\mathbf{x}$ varie), et nous devons également spécifier le nombre souhaité $\tilde{K}$ de coefficients sélectionnés. Le reste de l'algorithme (étapes 4 à 7) reste inchangé.

De toute évidence, l'un des points forts de cette variante de sélection est qu'elle peut être appliquée à n'importe quelle base fonctionnelle, sans besoin d'orthonormalisation. Cependant, elle ajoute un paramètre de réglage $\lambda(\mathbf{x})$ pour tout $\mathbf{x} \in \mathcal{X}$, ce qui augmente le coût de calcul global. Une alternative moins coûteuse est de considérer un paramètre de pénalité λ commun quel que soit $\mathbf{x} \in \mathcal{X}$. Cela suppose que les sorties $y_{\mathbf{x}}(.)$ ont le même niveau de régularité quand $\mathbf{x}$ varie. λ peut ensuite être calibré par validation croisée, en plus de $\tilde{K}$, dans l'algorithme 1.

Pour résumer, la variante sans orthonormalisation consiste à rajouter en entrée du modèle le paramètre de pénalisation λ, et à remplacer les étapes 1 à 3 de l'algorithme 1 par les deux étapes suivantes :

1. Décomposer les $(y_{\mathbf{x}^{(i)}}(\mathbf{z}))_{i=1,\dots,n}$ dans la base $\Phi(\mathbf{z})$ (voir (3.2)), en estimant les coefficients $(\alpha_k(\mathbf{x}^{(i)}))_{k=1,\dots,K}$ tel que soit résolu :

$$\underset{\alpha_1(x^{(i)}),\dots,\alpha_K(x^{(i)})}{min} ||y_{\mathbf{x}^{(i)}}(\mathbf{z}) - \sum_{k=1}^{K} \alpha_k(\mathbf{x}^{(i)})\phi_k(\mathbf{z})||^2 + \lambda \sum_{k=1}^{K} |\alpha_k(\mathbf{x}^{(i)})|, \ i = 1, \dots, n$$

   Contrairement à (3.5), λ ne dépend pas de $\mathbf{x}$.

2. Trier les coefficients $(\alpha_k(\mathbf{x}))_{k=1,\dots,K}$ selon l'ordre décroissant des $\mathbb{E}(\alpha_k(\mathbf{x}))$ ou $\mathbb{P}(\alpha_k(\mathbf{X}) \neq 0)$, $k = 1, \dots, K$. Puis, sélectionner les $\tilde{K}$ coefficients les plus importants.

3.2.2 Formulation de la loi de prédiction

D'après la propriété 3.1, pour une base $\Phi(.)$ non orthonormée, l'ACPF est équivalent à appliquer une ACP aux coefficients de la base orthonormalisée. Dans cette section, les propriétés sont donc formulées pour une base orthonormée. L'algorithme 1 nous permet de construire le modèle suivant pour $Y_{\mathbf{x}^\star}(.)$:

$$\begin{array}{rcl} Y_{\mathbf{x}^\star}(\mathbf{z}) & = & \sum_{k=1}^{\tilde{K}} \alpha_{(k)}(\mathbf{x}^\star)\phi_{(k)}(\mathbf{z}) + \sum_{k=\tilde{K}+1}^{K} \overline{\alpha}_{(k)}\phi_{(k)}(\mathbf{z}) \\ & = & \sum_{k=1}^{\tilde{K}} \left(\sum_{l=1}^{n_{PC}} t_l(\mathbf{x}^\star)\omega_{(k),l} \right) \phi_{(k)}(\mathbf{z}) + \sum_{k=\tilde{K}+1}^{K} \overline{\alpha}_{(k)}\phi_{(k)}(\mathbf{z}) \end{array} \quad (3.6)$$

avec $\overline{\alpha}_{(k)} = \mathbb{E}[\alpha_{(k)}(\mathbf{X})] \approx \frac{1}{n}\sum_{i=1}^{n}\alpha_{(k)}(\mathbf{x}_i)$. On définit $y_{\mathbf{x}^\star}(\mathbf{z})$ comme une réalisation de la variable aléatoire $Y_{\mathbf{x}^\star}(\mathbf{z})$. Sa prédiction est définie par la prédiction des coordonnées des coefficients $\boldsymbol{\alpha}_{\tilde{K}}(\mathbf{x}^\star) = (\alpha_{(1)}(\mathbf{x}^\star), \ldots, \alpha_{(\tilde{K})}(\mathbf{x}^\star))^\top$ dans la base de vecteurs propres $\mathbf{W} = (\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{n_{PC}})$: $\mathbf{t}(\mathbf{x}^\star) = (t_1(\mathbf{x}^\star), \ldots, t_{n_{PC}}(\mathbf{x}^\star))^\top$.

À l'étape 5 de l'algorithme 1, $\mathbf{t}(\mathbf{x}^\star)$ est prédit en utilisant des modèles de régression par processus gaussien indépendants pour chaque $t_l(\mathbf{x}^\star)$. Cela semble être une procédure raisonnable, puisque les scores (coordonnées dans la base de vecteurs propres) sont décorrélés en raison de l'application de l'ACP [Jolliffe, 2002]. On a donc $t_l(\mathbf{x}^\star) \sim \mathcal{N}(\hat{t}_l(\mathbf{x}^\star), \sigma_l^2(\mathbf{x}^*))$, avec $\hat{t}_l(\mathbf{x}^\star)$ la moyenne de krigeage et $\sigma_l^2(\mathbf{x}^*)$ la variance prédictive. $\mathbf{t}(\mathbf{x}^\star)$ est alors un vecteur gaussien de vecteur moyenne $\hat{\mathbf{t}}(\mathbf{x}^\star)$ et de matrice de covariance $\boldsymbol{\Sigma}(\mathbf{x}^\star)$, avec :

$$\hat{\mathbf{t}}(\mathbf{x}^\star) = (\hat{t}_1(\mathbf{x}^\star), \ldots, \hat{t}_{n_{PC}}(\mathbf{x}^\star))^\top \tag{3.7}$$

et

$$\boldsymbol{\Sigma}(\mathbf{x}^\star) = \begin{pmatrix} \sigma_1^2(\mathbf{x}^*) & & \\ & \ddots & \\ & & \sigma_{n_{PC}}^2(\mathbf{x}^*) \end{pmatrix} \tag{3.8}$$

D'après la définition 2.2, toute combinaison linéaire des éléments d'un vecteur gaussien suit une loi gaussienne. $\sum_k^{\tilde{K}}\left(\sum_{l=1}^{n_{PC}} t_l(\mathbf{x}^\star)\omega_{(k),l}\right)\phi_{(k)}(\mathbf{z})$ étant une combinaison linéaire des éléments de $\mathbf{t}(\mathbf{x}^\star)$ et $\sum_{k=\tilde{K}+1}^{K}\overline{\alpha}_{(k)}\phi_{(k)}(\mathbf{z})$ étant une constante, $Y_{\mathbf{x}^\star}(\mathbf{z})$ suit alors une loi gaussienne. Une loi gaussienne est entièrement définie par sa moyenne et sa variance. C'est pourquoi nous calculons celles de $Y_{\mathbf{x}^\star}(\mathbf{z})$.

L'équation 3.6 peut être simplifiée comme suit :

$$Y_{\mathbf{x}^\star}(\mathbf{z}) = \boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\mathbf{t}(\mathbf{x}^\star) + \overline{\boldsymbol{\alpha}}_{-\tilde{K}}^\top\boldsymbol{\phi}_{-\tilde{K}}(\mathbf{z}) \tag{3.9}$$

avec $\overline{\boldsymbol{\alpha}}_{-\tilde{K}} = (\overline{\alpha}_{(\tilde{K}+1)}, \ldots, \overline{\alpha}_{(K)})^\top$, $\boldsymbol{\phi}_{\tilde{K}} = (\phi_{(1)}, \ldots, \phi_{(\tilde{K})})^\top$ et $\boldsymbol{\phi}_{-\tilde{K}}(\mathbf{z}) = (\phi_{(\tilde{K}+1)}, \ldots, \phi_{(K)})^\top$. La moyenne et la variance de $Y_{\mathbf{x}^\star}(\mathbf{z})$ sont donc respectivement (3.10) et (3.11).

$$\begin{aligned} \hat{y}_{\mathbf{x}^\star}(\mathbf{z}) &= \mathbb{E}\left[Y_{\mathbf{x}^\star}(\mathbf{z})\right] \\ &= \mathbb{E}\left[\boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\mathbf{t}(\mathbf{x}^\star) + \overline{\boldsymbol{\alpha}}_{-\tilde{K}}^\top\boldsymbol{\phi}_{-\tilde{K}}(\mathbf{z})\right] \\ &= \boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\left(\mathbb{E}\left[\mathbf{t}(\mathbf{x}^\star)\right]\right) + \overline{\boldsymbol{\alpha}}_{-\tilde{K}}^\top\boldsymbol{\phi}_{-\tilde{K}}(\mathbf{z}) \\ &= \boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\hat{\mathbf{t}}(\mathbf{x}^\star) + \overline{\boldsymbol{\alpha}}_{-\tilde{K}}^\top\boldsymbol{\phi}_{-\tilde{K}}(\mathbf{z}) \end{aligned} \tag{3.10}$$

$$\begin{aligned} \sigma_{y_{\mathbf{x}^\star}}^2(\mathbf{z}) &= \mathbb{V}\mathrm{ar}\left[Y_{\mathbf{x}^\star}(\mathbf{z})\right] \\ &= \mathbb{V}\mathrm{ar}\left[\boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\mathbf{t}(\mathbf{x}^\star)\right] \\ &= \boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\left(\mathbb{V}\mathrm{ar}\left[\mathbf{t}(\mathbf{x}^\star)\right]\right)\mathbf{W}^\top\boldsymbol{\phi}_{\tilde{K}}(\mathbf{z}) \\ &= \boldsymbol{\phi}_{\tilde{K}}(\mathbf{z})^\top\mathbf{W}\boldsymbol{\Sigma}(\mathbf{x}^\star)\mathbf{W}^\top\boldsymbol{\phi}_{\tilde{K}}(\mathbf{z}) \end{aligned} \tag{3.11}$$

Finalement, on a $Y_{\mathbf{x}^\star}(\mathbf{z}) \sim \mathcal{N}(\hat{y}_{\mathbf{x}^\star}(\mathbf{z}), \sigma^2_{y_{\mathbf{x}^\star}}(\mathbf{z}))$. $\hat{y}_{\mathbf{x}^\star}(\mathbf{z})$ est considéré comme la prédiction de $y_{\mathbf{x}^\star}(\mathbf{z})$. $\sigma^2_{y_{\mathbf{x}^\star}}(\mathbf{z})$ est sa variance prédictive.

3.3 Application à un cas analytique

On note respectivement GP^{PCA}, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, et $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$ les méta-modèles effectués par ACP standard, par ACPF avec base d'ondelettes, et par ACPF avec base B-splines. Ils sont appliqués à un cas analytique, qui est présenté dans la section 3.3.1. La section 3.3.2 explique comment sont définies les bases d'ondelettes et B-splines. Les paramètres optimaux de l'ACP et de l'ACPF sont estimés par validation croisée. Ensuite les trois méta-modèles sont comparés dans la section 3.3.3. Toutes les implémentations ont été faites en utilisant le langage de programmation **R**.

Un package R, nommé GpOutput2D (voir chapitre 6), a été développé. GpOutput2D contient des fonctions pour appliquer l'ACPF et pour construire des modèles de régression GP aux données fonctionnelles à deux dimensions. Il est basé sur d'autres packages R pour la décomposition en ondelettes et B-splines, et pour les modèles de krigeage : waveslim, orthogonalsplinebasis, DiceKriging et kergp.

3.3.1 Description de la fonction Campbell2D

La fonction Campbell2D a huit entrées ($d = 8$) et une sortie spatiale en sortie (c'est-à-dire une fonction qui dépend de deux paramètres ($\mathbf{z} = (z_1, z_2)$), correspondant aux coordonnées spatiales).

$$\begin{array}{rccc} f & : & [-1,5]^8 & \to & \mathbb{L}^2([-90,90]^2) \\ & & \mathbf{x} = (x_1, \ldots, x_8) & \mapsto & y_{\mathbf{x}}(\mathbf{z}) \end{array} \tag{3.12}$$

où $\mathbf{z} = (z_1, z_2) \in [-90, 90]^2$, $x_j \in [-1, 5]$ for $j = 1, \ldots, 8$, et

$$\begin{aligned} y_{\mathbf{x}}(z_1, z_2) = \; & x_1 \exp\left[-\tfrac{(0.8z_1+0.2z_2-10x_2)^2}{60x_1^2}\right] + (x_2 + x_4) \exp\left[\tfrac{(0.5z_1+0.5z_2)x_1}{500}\right] + \\ & x_5(x_3 - 2) \exp\left[-\tfrac{(0.4z_1+0.6z_2-20x_6)^2}{40x_5^2}\right] + \\ & (x_6 + x_8) \exp\left[\tfrac{(0.3z_1+0.7z_2)x_7}{250}\right] \end{aligned} \tag{3.13}$$

La figure 3.1 montre des exemples de sorties de Campbell2D. La carte de sortie présente de fortes hétérogénéités spatiales, parfois avec des limites nettes. De plus, la distribution spatiale est différente selon les valeurs de $\mathbf{x}$. Un échantillon d'apprentissage de taille $n = 200$ est considéré avec un plan d'expérience construit en utilisant un échantillonnage par hypercube latin (appelé LHS) optimisé par l'algorithme SA [Dupuy et al., 2015], implémenté dans le package *DiceDesign* du logiciel **R** . Les points du plan d'expérience et les cartes de sorties associées sont respectivement notés $\mathbf{x}^{(i)}$ et $y_i(\mathbf{z})$ $(= y_{\mathbf{x}^{(i)}}(\mathbf{z}))$, $i = 1 \ldots, n$. Pour l'application, le domaine spatial $[-90, 90]^2$ est discrétisé en une grille

uniforme de dimension 64×64. On remarque que les deux dimensions doivent être une puissance de deux, une exigence de la décomposition en ondelettes.

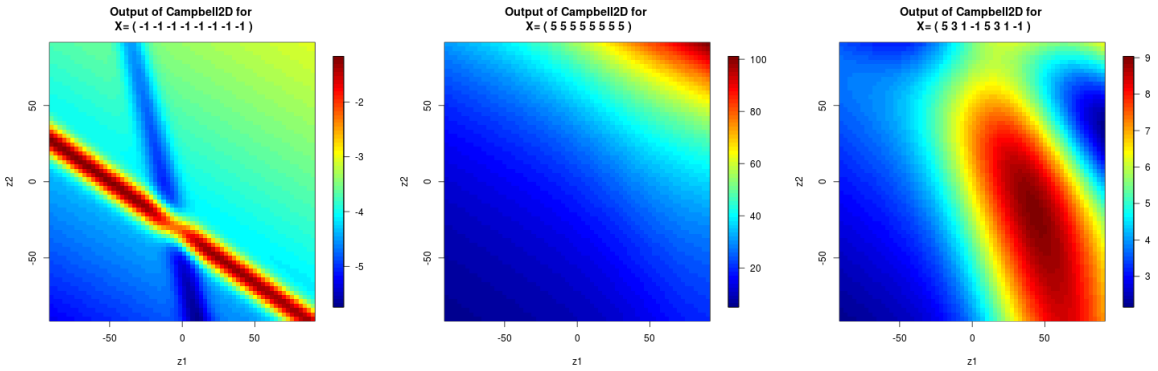

FIGURE 3.1 – Exemple de sorties de la fonction Campbell2D. De gauche à droite, $\mathbf{x} = (-1,-1,-1,-1,-1,-1,-1,-1)$, $\mathbf{x} = (5,5,5,5,5,5,5,5)$, and $\mathbf{x} = (5,3,1,-1,5,3,1,-1)$.

3.3.2 Calibration de l'ACPF

Pour la décomposition sur base d'ondelettes, les ondelettes de Daubechies D4 [Daubechies, 1988] sont utilisées, afin d'avoir un compromis entre la taille du support et le domaine fréquentiel. La dimension de la base est la même que le domaine spatial : ici, $K = 64 \times 64 = 4\ 096$. Néanmoins, l'approximation multi-résolution de la sortie spatiale a besoin de définir un nombre de résolutions (aussi appelé niveau de décomposition) [Mallat, 1999], que l'on note J. Pour les B-splines, des splines de degré 1 sont considérées, et des nœuds sont choisis également espacés. Pour la simplicité, le même nombre de nœuds est considéré pour les deux dimensions. Les bases d'ondelettes et B-splines sont sélectionnées telles que la racine de la moyenne de l'erreur quadratique de l'approximation des cartes de l'échantillon d'apprentissage soit minimisée :

$$\mathrm{RMSE}_{\Phi}(\mathbf{z}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i(\mathbf{z}) - \hat{y}_i^{\Phi}(\mathbf{z}))^2} \tag{3.14}$$

avec $\hat{y}_i^{\Phi}(.)$ l'approximation de $y_i(.)$ dans la base $\Phi(.)$, qui est, ici, la base d'ondelettes ou B-splines. Pour la suite, on pose $J = 1$ la profondeur de la décomposition en ondelettes. Le RMSE de la décomposition sur une base B-splines orthonormalisée (voir la figure 3.2) converge vers 0 à partir de $K = 35^2$. On choisit donc $K = 35^2$, pour la méthode utilisant les splines.

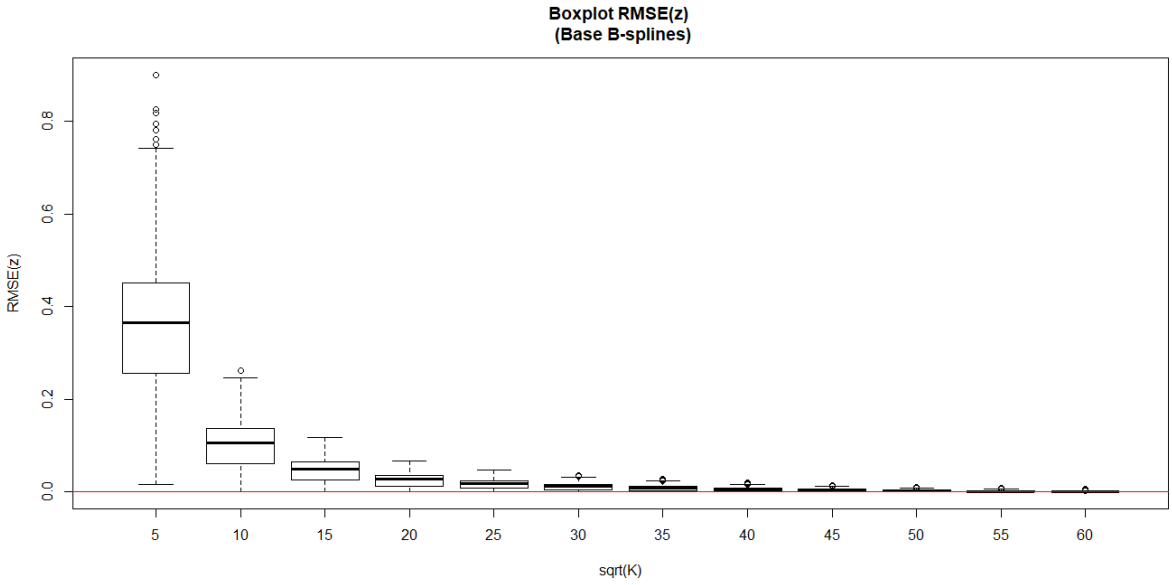

Figure 3.2 – Boxplot des cartes RMSE mesurant l'erreur d'approximation sur la base B-splines, en fonction du nombre de nœuds pour chaque dimension, également espacés sur $[-90, 90]$.

Après avoir ajusté les bases d'ondelettes et de B-splines selon les cartes observées de l'échantillon d'apprentissage, le nombre de coefficients à prédire par krigeage ($\tilde{K}$) et le nombre de composantes principales (n_{PC}) doivent être calibrés tels que l'erreur de prédiction soit minimisée. Une validation croisée à k blocs est donc appliquée [Hastie et al., 2009], avec $k = 10$. L'échantillon d'apprentissage est divisé en dix sous-échantillons, générés aléatoirement. Les cartes d'un des sous-échantillons sont supposées inconnues et prédites selon l'algorithme 1, en se basant sur les observations des autres sous-échantillons. Ceci est réitéré pour chaque sous-échantillon. Afin d'évaluer la performance prédictive du méta-modèle, la racine de l'erreur quadratique moyenne (ici, appelée RMSE) est calculée pour chaque sous-échantillon de la validation croisée :

$$\mathrm{RMSE}_l(\mathbf{z}) = \sqrt{\frac{1}{n_l}\sum_{i'=1}^{n_l}\left(y_{\mathbf{x}_{i'}^{(l)}}(\mathbf{z}) - \hat{y}_{\mathbf{x}_{i'}^{(l)}}(\mathbf{z})\right)^2}, \ \forall l \in \{1, \ldots, k\} \tag{3.15}$$

où $(\mathbf{x}_{i'}^{(l)}, y_{\mathbf{x}_{i'}^{(l)}}(\mathbf{z}))$ est la i-ème observation (entrées, sortie) du l-ème sous-échantillon de taille $n_l = \frac{n}{k} = 20$, et $\hat{y}_{\mathbf{x}_{i'}^{(l)}}(\mathbf{z})$ est l'estimation de $y_{\mathbf{x}_{i'}^{(l)}}(\mathbf{z})$. Ensuite, un RMSE global de la validation croisée est calculé en moyennant les RMSEs des sous-échantillons :

$$\mathrm{RMSE}_{\mathrm{CV}}(\mathbf{z}) = \frac{1}{k}\sum_{l=1}^{k}\mathrm{RMSE}_l(\mathbf{z}) \tag{3.16}$$

Pour une validation croisée, on obtient donc une carte d'erreurs locales. Pour choisir $\tilde{K}$ et n_{PC}, on applique une validation croisée pour plusieurs valeurs possibles de $\tilde{K}$ et n_{PC}. On choisit les plus petites valeurs de ($\tilde{K}$,n_{PC}) telles que l'erreur prédictive soit minimisée. Cependant, cela est difficilement faisable car l'erreur dépend de $\mathbf{z}$. Pour quantifier l'erreur locale en une quantité scalaire, on utilise le quantile 90% de (3.16) selon $\mathbf{z}$. La Figure 3.3 montre la valeur des quantiles selon $\tilde{K}$ et n_{PC}. $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$ et $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-spline}}$ désignent respectivement la méthodologie de l'algorithme 1 en utilisant les ondelettes et les B-splines.

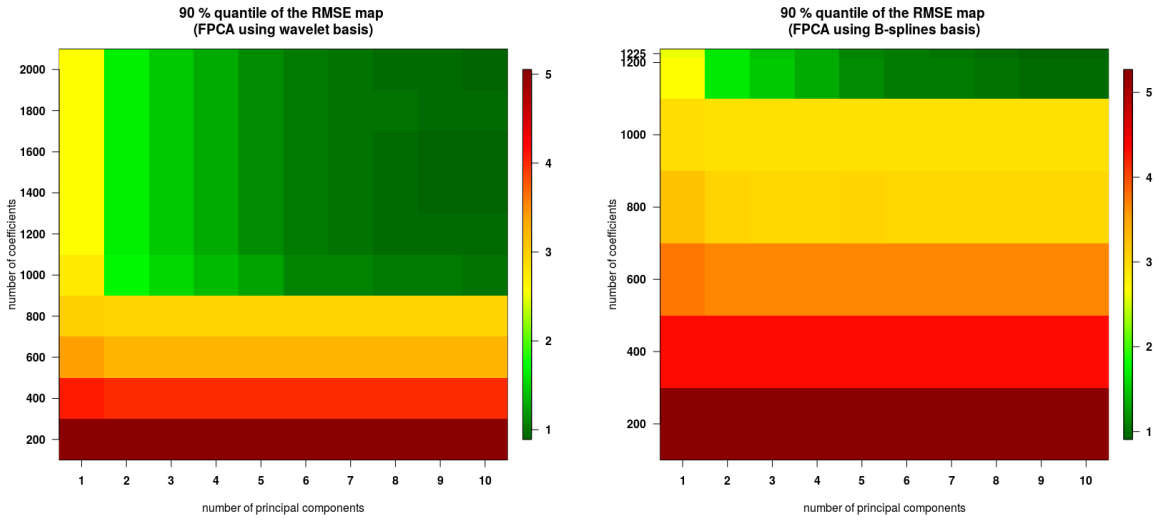

FIGURE 3.3 – Quantile 90% du RMSE de la validation croisée à 10 blocs : $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$(à gauche), $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$(à droite).

Pour $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$, une convergence du RMSE est observée à partir de $\tilde{K} = 1200$, quelle que soit la valeur de n_{PC}. Pour $\tilde{K} = 1200$, une convergence commence à $n_{PC} = 8$. Cependant, un RMSE plus petit est observé à $n_{PC} = 5$. À partir de la 6-ème composante principale, le pourcentage de variance expliquée est inférieur à 1%, qui est négligeable. Afin d'éviter le sur-apprentissage, on choisit $\tilde{K} = 1200$ and $n_{PC} = 5$ pour la méthode $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$. Pour $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$, quel que soit n_{PC}, la valeur du RMSE est minime pour $\tilde{K} = 1225 = 35^2$, qui correspond à la dimension totale de la base B-splines. Même si ce nombre est large, il reste raisonnable pour appliquer l'ACP, et est donc choisi pour l'ACPF. Finalement, pour les mêmes raisons que $\mathrm{GP}^{\mathrm{FPCA}}_{\text{wavelet}}$, on choisit aussi $n_{PC} = 5$ composantes principales pour $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$.

Pour la comparaison, on modélise $n_{PC} = 5$ composantes principales pour $\mathrm{GP}^{\mathrm{PCA}}$. Les cinq premières composantes principales correspondent à 98% de l'inertie totale pour les trois méthodes. Pour $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$, tous les coefficients de la base sont utilisés pour l'étape de l'ACP de l'algorithme 1, ce qui correspond à 100% de l'énergie moyenne (3.4). Pour

$\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, 29.3% des coefficients d'ondelette ($\tilde{K} = 1\,200$) sont gardés, ce qui correspond à presque 100% de l'énergie moyenne.

3.3.3 Précision de la prédiction

Dans cette section, un échantillon test de $n_{test} = 1000$ simulations de f est construit. Les entrées $\mathbf{x}$ sont tirées aléatoirement indépendamment selon une loi uniforme sur $[-1, 5]^8$. Les cartes de sorties sont supposées inconnues. Elles sont prédites par GP^{PCA}, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, ou $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, en utilisant les paramètres qui ont été calibrés dans la section 3.3.2, et en se basant sur les $n = 200$ observations de l'échantillon d'apprentissage. Pour comparer, on utilise $n_{PC} = 5$ composantes principales pour GP^{PCA}.

La racine de l'erreur quadratique moyenne (Eq.3.17), de chacune des méthodes est comparée dans la figure 3.4.

$$\text{RMSE}(\mathbf{z}) = \sqrt{\frac{1}{n_{test}} \sum_{i'=1}^{n_{test}} \left[y_{\mathbf{x}^{(i')}}(\mathbf{z}) - \hat{y}_{\mathbf{x}^{(i')}}(\mathbf{z}) \right]^2}, \quad \mathbf{z} \in [-90, 90]^2 \tag{3.17}$$

où $y_{\mathbf{x}^{(i')}}(\mathbf{z})$ et $\hat{y}_{\mathbf{x}^{(i')}}(\mathbf{z})$ sont respectivement la carte de sortie réelle et celle prédite pour $\mathbf{x}^{(i')}$, avec $i' = 1, \ldots, n_{test}$.

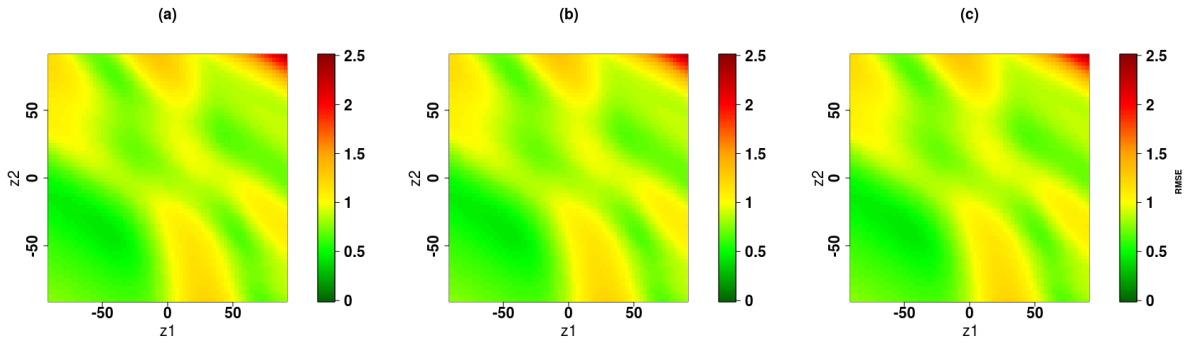

FIGURE 3.4 – Cartes RMSE obtenues avec $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, and GP^{PCA}, respectivement notées (a), (b), et (c).

On peut voir que les trois méthodes ont la même précision de prédiction. La précision de la prédiction peut aussi être quantifiée par un autre critère : le critère Q^2. Une version généralisée du critère (voir (3.18)) est définie.

$$Q^2 = 1 - \frac{\mathbb{E}_{\mathbf{z}} \left\{ \mathbb{E}_{\mathbf{X}} \left[\left(Y_{\mathbf{X}}(\mathbf{z}) - \hat{Y}_{\mathbf{X}}(\mathbf{z}) \right)^2 \right] \right\}}{\mathbb{E} \left[\mathbb{V}\text{ar} \left[Y_{\mathbf{X}}(\mathbf{z}) \right] \right]}. \tag{3.18}$$

En pratique, l'espérance est remplacée par la moyenne empirique. Le critère Q^2 compare l'erreur quadratique moyenne (que l'on appellera MSE) d'un modèle par rapport à la variance des observations, moyennées spatialement. Ici, GP^{PCA}, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, et $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$ont le même coefficient de prédiction, $Q^2 \approx 96.6\%$, qui est un score très satisfaisant.

On peut conclure que les trois modèles sont aussi efficaces, ce qui semble prometteur pour l'approche basée sur l'ACP fonctionnelle, qui réduit significativement le problème de dimension. En effet, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$utilise seulement 29.3% des coefficients d'ondelettes. $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$réduit la dimension à 1 225 au lieu de 4 096.

3.3.4 Variante sans orthonormalisation

La version de l'ACPF sans orthonormalisation a aussi été appliquée au cas analytique. L'étape de sélection des coefficients de l'algorithme 1 est modifiée en utilisant des modèles de régression Lasso pour estimer les coefficients de la décomposition sur la base B-splines, comme décrit dans la section 3.2.1.2. Dans un premier temps, nous étudions comment les coefficients de la base sont estimés pour une carte. Ensuite, les cartes de l'échantillon test de la section 3.3.3 sont estimées. La précision des prédictions est comparée à celles de $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$.

3.3.4.1 Exemple de régression Lasso sur une carte

Afin d'étudier l'estimation des coefficients de la base B-splines, une carte d'exemple est générée avec la fonction Campbell2D (aussi de dimension 64×64), avec pour vecteur d'entrées $\mathbf{x}' = (-1, -1, -1, -1, -1, -1, -1, -1)$ (voir la figure 3.7).

On construit une base B-splines de dimension $K = 35 \times 35 = 1225$ d'ordre 1. Le package **R** glmnet permet de construire des modèles de régression Lasso. On estime les coefficients dans la base B-splines telle que l'équation (3.5) soit résolue. Les valeurs des coefficients dépendent du paramètre de pénalisation $\lambda(\mathbf{x}')$. On représente dans la figure 3.5 l'évolution des coefficients B-splines en fonction de la valeur de $\lambda(\mathbf{x}')$. On remarque que plus $\lambda(\mathbf{x}')$ est grand et plus il y a de coefficients estimés à zéro. Il faut donc calibrer $\lambda(\mathbf{x}')$.

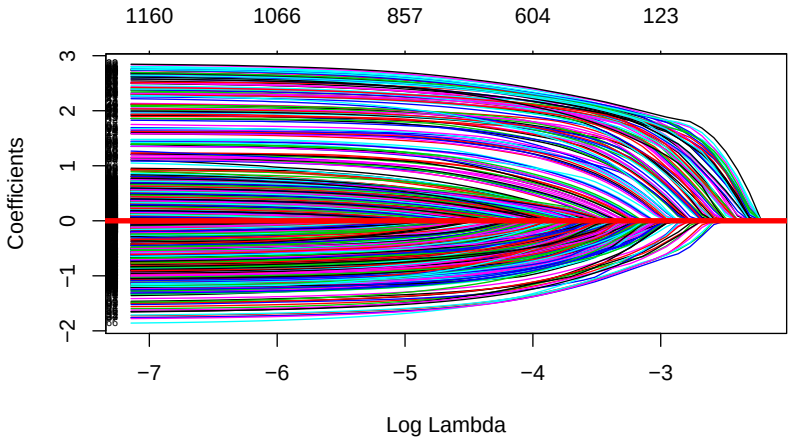

Figure 3.5 – Valeurs des coefficients dans la base B-splines en fonction de $\lambda(\mathbf{x}')$ (en échelle log). Le troisième axe, en haut de la figure, représente le nombre de coefficients non égaux à zéro.

En pratique, les coefficients et $\lambda(\mathbf{x}')$ sont estimés par validation croisée. La figure 3.6 représente l'erreur quadratique moyenne de la validation croisée 10−fold. On peut choisir $\lambda(\mathbf{x}')$ en fonction de deux critères : l'erreur quadratique moyenne et le nombre de coefficients non estimés à zéro. Ici, on choisit la plus grande valeur $\lambda(\mathbf{x}')$ telle que l'erreur quadratique moyenne soit minimisée : $\hat{\lambda}(\mathbf{x}') = 0.001$, soit $\tilde{K} = 1\ 160$ coefficients non estimés à zéro (voir la figure 3.5).

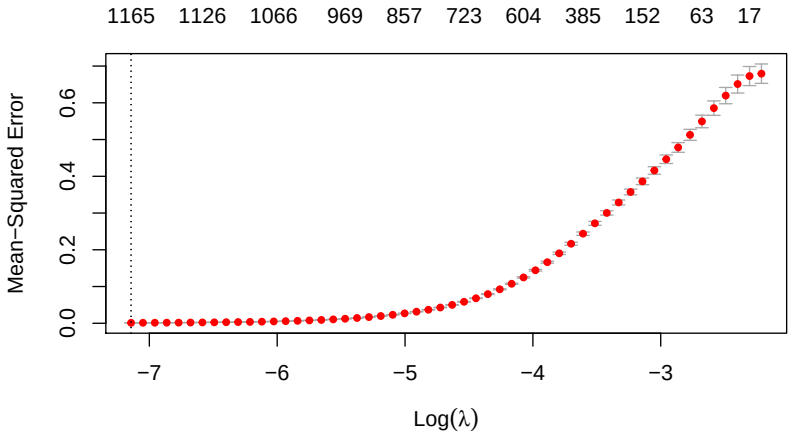

Figure 3.6 – Erreur quadratique moyenne de la validation croisée 10−fold en fonction de $\lambda(\mathbf{x}')$ (en échelle log).

La figure 3.7 compare la carte réelle avec la carte estimée par régression Lasso. Néanmoins, aucune différence n'est visible. C'est pourquoi nous regardons la différence entre les deux cartes et traçons sa valeur absolue dans la figure 3.8. Les plus grandes différences se situent au trait diagonal rouge (voir la figure 3.7), là où se situent les coefficients de plus grandes valeurs. On constate une irrégularité de l'erreur spatiale aux diagonales qui correspondent aux diagonales rouge et bleu de la figure 3.7. Néanmoins, l'erreur maximale est de 0.14 au maximum, ce qui est une erreur raisonnable par rapport à l'amplitude des valeurs de la carte.

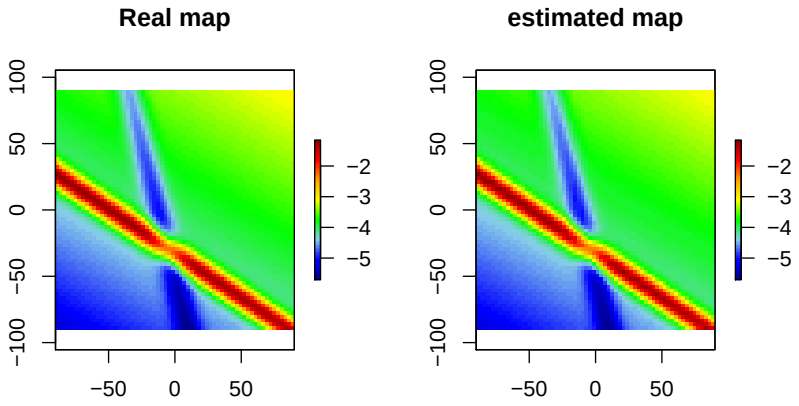

FIGURE 3.7 – À gauche, la sortie de Campbell2D pour le vecteur d'entrées $\mathbf{x}' = (-1, -1, -1, -1, -1, -1, -1, -1)$. À droite, la même carte estimée dans la base B-splines avec les coefficients estimées par un modèle de régression Lasso, avec $\lambda(\mathbf{x}') = 0.001$.

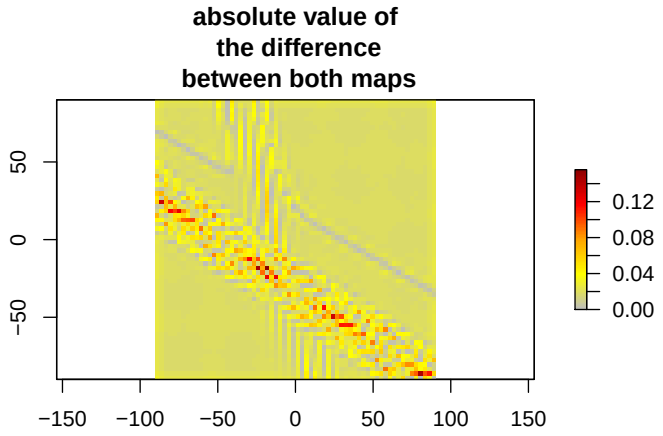

FIGURE 3.8 – Valeur absolue de la différence des cartes de la figure 3.7.

3.3.4.2 Prédiction avec sélection par régression Lasso

Nous commençons par calibrer les paramètres de l'ACPF. Ils sont choisis en appliquant la même procédure que dans la section 3.3.2. En plus de $\tilde{K}$ et n_{PC}, le paramètre de pénalisation λ (voir (3.5)) doit aussi être calibré. On considère un λ commun pour tout $\mathbf{x}$, afin d'économiser le temps de calcul. La figure 3.9 montre le quantile 90% des cartes RMSE de la validation croisée à 10 blocs, selon λ, $\tilde{K}$ et n_{PC}. Moins de valeurs de $\tilde{K}$ ont été testées en raison du temps de calcul des $n = 200$ modèles de régression Lasso, et de la multiplication entre la matrice de Gram et les coefficients de la base.

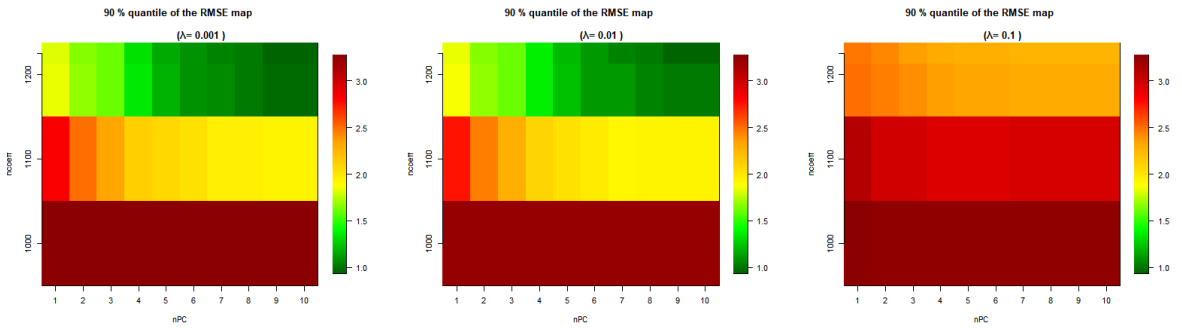

Figure 3.9 – Quantile 90% du RMSE de la validation croisée à 10 blocs. De gauche à droite $\lambda = 0.001,\ 0.01,\ 0.1$. $\tilde{K} = (1\,000, 1\,100, 1\,200, 1\,225)$ et $n_{PC} = (1, 2, \ldots, 10)$.

$\lambda = 0.1$ correspond à la figure avec les plus faibles valeurs de RMSE. Sinon, des résultats similaires sont observées pour $\lambda = 0.001$ et $\lambda = 0.01$. On choisit $\lambda = 0.01$, car plus de coefficients seront estimés à 0. Néanmoins, résultant des valeurs de RMSE les plus faibles, on choisit $\tilde{K} = K = 1\,225$, la taille totale de la base B-splines. En effet, en appliquant une régression Lasso pour chaque carte séparément, les coefficients estimés à zéro ne sont pas les mêmes d'une carte à une autre. On pose $n_{PC} = 5$ pour éviter le sur-apprentissage. À partir de la 6ème composante principale, les composantes représentent moins de 1% de l'inertie totale, ce qui est négligeable. On remarque que $\tilde{K}$ et n_{PC} sont les mêmes valeurs qui ont été choisies lors de la calibration $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$.

La méthode a été utilisée pour prédire les $n_{test} = 1\,000$ cartes de l'échantillon test. La carte RMSE a été calculée et comparée à celle obtenue avec $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$ dans la Figure 3.10. On peut voir que $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$, construit avec la version principale de l'algorithme 1, reste meilleur en terme de précision.

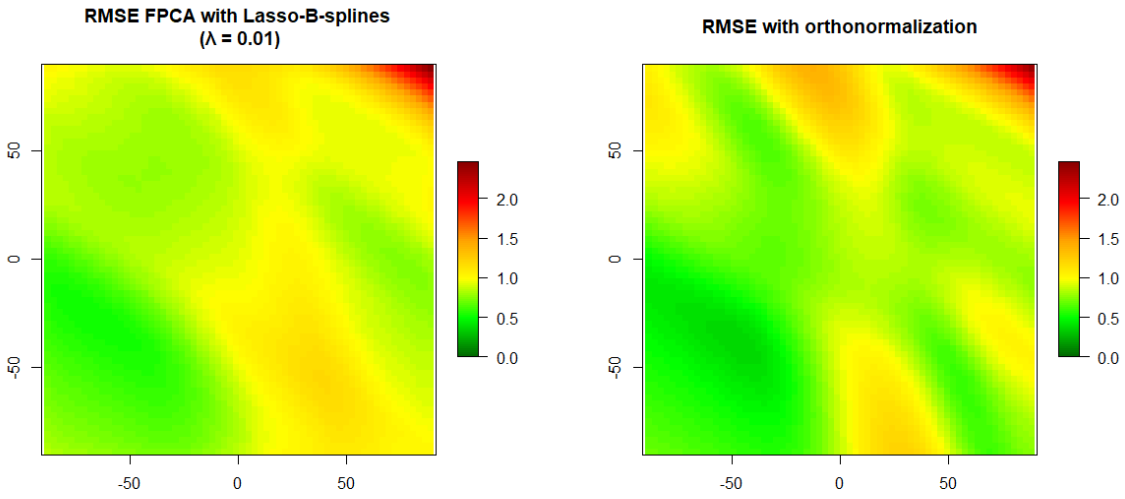

FIGURE 3.10 – Cartes RMSE : à gauche, sans orthonormalisation en utilisant la régression Lasso, à droite, en utilisant $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$.

Les codes des deux ACPF ont été exécutés avec un seul cœur d'un AMD Ryzen$^{\mathrm{TM}}$ 7 4700U CPU. Le temps de calcul de la variante basée sur Lasso est supérieur à une minute, comparé à moins d'une seconde pour l'ACPF avec l'orthonormalisation des B-splines. En raison des dimensions des cartes du cas d'étude des submersions marines, cette variante sans orthonormalisation est difficilement traitable. C'est pourquoi, on ne l'appliquera pas dans la partie III.

Chapitre 4

Analyse de sensibilité pour les modèles avec sortie spatiale

Sommaire

4.1 Introduction

Dans cette section, on considère aussi le simulateur f comme défini dans la section 3.1.1.

$$\begin{array}{rccc} f & : & \mathcal{X} \subseteq \mathbb{R}^d & \rightarrow & \mathbb{L}^2(\mathcal{Z}) \\ & & \mathbf{x} & \mapsto & y_{\mathbf{x}}(\mathbf{z}) \end{array}$$

où $\mathbf{x}$ est le vecteur des entrées, $\mathcal{Z}$ est le domaine spatial, et $y_{\mathbf{x}}(\mathbf{z})$ est la valeur de la sortie spatiale à la localisation $\mathbf{z} \in \mathcal{Z}$.

Nous cherchons à analyser la sensibilité de la sortie $y_{\mathbf{x}}(.)$ à l'entrée $\mathbf{x}$, sur l'ensemble du domaine spatial $\mathcal{Z}$. Plusieurs méthodes d'analyse de sensibilité existent comme les coefficients de corrélations, ou la méthode de Morris [Faivre et al., 2016] [Iooss, 2011, Iooss and Lemaître, 2015], qui sont non coûteux à évaluer. Cependant ces méthodes ne quantifient pas les interactions et les non-linéarités. Nous nous intéressons donc aux indices de sensibilité basés sur la décomposition de la variance, appelés indices de Sobol [Sobol, 1993, Saltelli et al., 2008]. Cependant, la sortie de f est fonctionnelle.

Plusieurs méthodes ont été développées pour les modèles avec sortie fonctionnelle. [Campbell et al., 2006] propose de réduire la dimension de la sortie fonctionnelle, en utilisant une base de fonctions adaptée (base de Fourier, ACP etc.). L'analyse de sensibilité est ensuite faite dans la base projetée. Un package **R**, appelé « multisensi » (https://cran.r-project.org/web/packages/multisensi/vignettes/multisensi-vignette.pdf), existe pour l'analyse de sensibilité de modèle avec sortie multivariée. Mais il n'est pas adapté pour le cas où le modèle a une sortie spatiale. L'utilisation d'une base de fonctions adaptée pour les données spatiales a été proposé par [Marrel et al., 2011] : la base d'ondelettes, qui permet d'étudier la sortie spatiale à la fois dans le domaine fréquentiel et dans le domaine spatial [Mallat, 1999]. La décomposition de la variance de chaque coefficient d'ondelettes permet de reconstituer des cartes d'indices de Sobol pour chaque variable d'entrée et interaction. Pour d grand, le nombre de cartes à analyser peut être difficilement gérable. C'est pourquoi, on s'intéresse à un indice de sensibilité généralisant l'influence de $\mathbf{x}$ sur $y_{\mathbf{x}}(.)$, sur l'ensemble du domaine $\mathcal{Z}$. [Lamboni et al., 2011] propose un indice basé sur la trace de la matrice de covariance de $y_{\mathbf{x}}(\mathbf{z})$, selon $\mathbf{z}$. En réduisant la dimension de $\mathcal{Z}$ par ACP ou ACPF, il est équivalent à la moyenne des indices de Sobol des scores, pondérée par les valeurs propres. Dans ce chapitre, on propose aussi de généraliser le calcul de l'indice de [Lamboni et al., 2011], pour une base de fonctions quelconque.

Dans un premier temps, on présente l'indice de [Lamboni et al., 2011]. Ensuite, le calcul de cet indice est généralisé pour une base de fonctions quelconque. Enfin, les analyses de sensibilité spatiale et généralisée de la fonction Campbell2D (voir la section 3.3.1) sont faites.

4.2 Indice de sensibilité généralisé

4.2.1 Indice de Lamboni

On appelle GSI, l'indice généralisé de [Lamboni et al., 2011] mesurant l'influence des entrées sur l'ensemble de la sortie spatiale. [Gamboa et al., 2020] a ajouté des arguments théoriques afin de confirmer sa définition.

Définition 4.1. *L'indice de sensibilité spatial généralisé de* $y_{\mathbf{x}}(\mathbf{z})$ *par rapport à* $\mathbf{x}_\omega$*, avec* $\omega \subseteq \{1, \ldots, d\}$*, est :*

$$\mathrm{GSI}_\omega = \frac{\mathrm{Trace}\left(\mathbb{C}\mathrm{ov}\left(\mathbb{E}_{\mathbf{X}}[y_{\mathbf{X}}(\mathbf{z})|\mathbf{X}_\omega]\right)\right)}{\mathrm{Trace}\left(\mathbb{C}\mathrm{ov}\left(y_{\mathbf{X}}(\mathbf{z})\right)\right)} \tag{4.1}$$

avec $\mathrm{Trace}\left(\mathbb{C}\mathrm{ov}\left(y_{\mathbf{X}}(\mathbf{z})\right)\right) = \int_{\mathcal{Z}} \mathbb{V}\mathrm{ar}(y_{\mathbf{X}}(\mathbf{z}))d\mu(\mathbf{z})$ *(avec une définition similaire pour le numérateur). L'indice de sensibilité spatial généralisé total de la variable* $\mathbf{X}_j$*,* $j \in \{1, \ldots, d\}$*, est* $\mathrm{GTSI}_j = \sum_{\omega, j\in\omega} \mathrm{GSI}_\omega$*.*

La sortie de f est de dimension infinie, ce qui rend difficile une analyse directe. C'est pourquoi, on a besoin de réduire la dimension. Dans ce but, une ACPF est appliquée à

$y_{\mathbf{x}}(.)$ (voir la section 2.3) dans [Lamboni et al., 2011]. La décomposition suivante est ainsi obtenue :

$$y_{\mathbf{x}}(\mathbf{z}) = \mu(\mathbf{z}) + \sum_{l=1}^{n_{PC}} \theta_l(\mathbf{x})\xi_l(\mathbf{z}) \tag{4.2}$$

avec $\mu(\mathbf{z}) = \mathbb{E}(Y_{\mathbf{X}}(\mathbf{z}))$, et $(\theta_l(\mathbf{x}))_{l=1,\dots,n_{PC}}$ sont les coefficients (scores) de $y_{\mathbf{x}}(\mathbf{z})$ dans la base des fonctions propres $(\xi_l(\mathbf{z}))_{l=1,\dots,n_{PC}}$. Ensuite, GSI peut être calculé grâce à la propriété 4.1.

Propriété 4.1. *[Lamboni et al., 2011] Pour tout $\omega \subseteq \{1,\dots,d\}$, GSI_ω peut être calculé de la manière suivante :*

$$\mathrm{GSI}_\omega = \frac{\sum_{l=1}^{n_{PC}} \lambda_l \mathrm{SI}_{\omega,l}}{\sum_{l=1}^{n_{PC}} \lambda_l} \tag{4.3}$$

où λ_l est la $l-$ème valeur propre, $\mathrm{SI}_{\omega,l}$ est l'indice de Sobol de la $l-$ème composante principale, ce qui correspond à l'influence de $\mathbf{x}_\omega$ sur la valeur de θ_l. De plus, $0 \leq \mathrm{GSI}_\omega \leq 1$ et $\sum_{\omega\subseteq\{1,\dots,d\}} \mathrm{GSI}_\omega = 1$.

4.2.2 Extension de l'indice

Dans cette section, on présente la propriété 4.2, qui est une généralisation du calcul de GSI, pour une base quelconque de fonctions utilisée pour approximer $y_{\mathbf{x}}(\mathbf{z})$.

Propriété 4.2. *Soit $\phi = (\phi_1,\dots,\phi_K)^\top$ une base de fonctions, avec pour matrice de Gram $\mathbf{G} = \int_{\mathcal{Z}} \phi(\mathbf{z})\phi(\mathbf{z})^\top d\mu(\mathbf{z})$. On suppose que $y_{\mathbf{x}}$ est décomposé dans ϕ, comme suit :*

$$y_{\mathbf{X}}(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k(\mathbf{X})\phi_k(\mathbf{z}) \tag{4.4}$$

En notant $\alpha(\mathbf{X}) = (\alpha_1(\mathbf{X}),\dots,\alpha_K(\mathbf{X}))^\top$ le vecteur des coefficients, GSI de y peut être calculé comme suit :

$$\mathrm{GSI}_\omega = \frac{\mathrm{Trace}(\mathbb{C}\mathrm{ov}(\mathbb{E}[\alpha(\mathbf{X})|\mathbf{X}_\omega])\mathbf{G})}{\mathrm{Trace}(\mathbb{C}\mathrm{ov}(\alpha(\mathbf{X}))\mathbf{G})} \tag{4.5}$$

Preuve D'après la définition 4.1, on a :

$$\mathrm{GSI}_\omega = \frac{\mathrm{Trace}\left(\mathbb{C}\mathrm{ov}\left(\mathbb{E}_{\mathbf{X}}[y_{\mathbf{X}}(\mathbf{z})|\mathbf{X}_\omega]\right)\right)}{\mathrm{Trace}\left(\mathbb{C}\mathrm{ov}\left(y_{\mathbf{X}}(\mathbf{z})\right)\right)} = \frac{\int_{\mathcal{Z}} \mathbb{V}\mathrm{ar}(\mathbb{E}[y_{\mathbf{X}}(\mathbf{z})|\mathbf{X}_\omega])d\mu(\mathbf{z})}{\int_{\mathcal{Z}} \mathbb{V}\mathrm{ar}(y_{\mathbf{X}}(\mathbf{z}))d\mu(\mathbf{z})}$$

Après décomposition de $y_{\mathbf{x}}$ dans la base ϕ (voir (4.4)), on a :

$$\mathbb{V}\mathrm{ar}(y_{\mathbf{X}}(\mathbf{z})) = \sum_{k,k'=1}^{K} (\mathbb{C}\mathrm{ov}(\alpha(\mathbf{X})))_{k,k'}\phi_k(\mathbf{z})\phi_{k'}(\mathbf{z})$$

et par linéarité de l'espérance :

$$\mathbb{V}\mathrm{ar}(\mathbb{E}[y_{\mathbf{X}}(\mathbf{z})|\mathbf{X}_\omega]) = \sum_{k,k'=1}^{K} (\mathbb{C}\mathrm{ov}(\mathbb{E}[\alpha(\mathbf{X})|\mathbf{X}_\omega]))_{k,k'}\phi_k(\mathbf{z})\phi_{k'}(\mathbf{z})$$

On obtient donc :

$$\mathrm{GSI}_\omega = \frac{\sum_{k,k'=1}^{K}(\mathbb{C}\mathrm{ov}(\mathbb{E}[\alpha(\mathbf{X})|\mathbf{X}_\omega]))_{k,k'}\int_{\mathcal{Z}}\phi_k(\mathbf{z})\phi_{k'}(\mathbf{z})d\mu(\mathbf{z})}{\sum_{k,k'=1}^{K}(\mathbb{C}\mathrm{ov}(\alpha(\mathbf{X})))_{k,k'}\int_{\mathcal{Z}}\phi_k(\mathbf{z})\phi_{k'}(\mathbf{z})d\mu(\mathbf{z})}$$

Finalement, avec $\mathbf{G} = \int_{\mathcal{Z}}\phi(\mathbf{z})\phi(\mathbf{z})^\top d\mu(\mathbf{z})$,

$$\mathrm{GSI}_\omega = \frac{\sum_{k,k'=1}^{K}(\mathbb{C}\mathrm{ov}(\mathbb{E}[\alpha(\mathbf{X})|\mathbf{X}_\omega]))_{k,k'}G_{k,k'}}{\sum_{k,k'=1}^{K}(\mathbb{C}\mathrm{ov}(\alpha(\mathbf{X})))_{k,k'}G_{k,k'}} = \frac{\mathrm{Trace}(\mathbb{C}\mathrm{ov}(\mathbb{E}[\alpha(\mathbf{X})|\mathbf{X}_\omega])\mathbf{G})}{\mathrm{Trace}(\mathbb{C}\mathrm{ov}(\alpha(\mathbf{X}))\mathbf{G})}$$

La dernière égalité est expliquée par la propriété suivante : $\mathrm{Trace}(AB^\top) = \sum_{k,l}A_{k,l}B_{k,l}$, pour toute matrice A et B. □

4.3 Analyse de sensibilité de la fonction Campbell2D

Dans cette section, nous procédons à l'analyse de sensibilité de la fonction Campbell2D, présentée dans la section 3.3.1. Dans la section 3.3.3, les méta-modèles sont tous les trois de même précision. Les observations de l'analyse de sensibilité sont obtenues en utilisant $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$. Le méta-modèle est construit à partir de l'échantillon d'apprentissage de taille $n = 200$ qui a été généré dans la section 3.3.1. Les indices de Sobol des scores (coordonnées dans la base des fonctions propres) sont estimés en utilisant la méthode de [Saltelli, 2002], qui dépend de deux échantillons initiaux. La fonction sobolSalt, du package **R** « sensitivity », a été utilisée. Par conséquent, deux jeux d'échantillons d'entrée de taille $n_0 = 10^4$ sont générés aléatoirement, pour un total de $n_0(d+2) = 10^5$ simulations de f. Les échantillons initiaux sont des échantillons d'hypercube latin, qui sont construits aléatoirement à partir d'une distribution uniforme.

Nous commençons par une analyse de sensibilité spatiale, faite à partir des indices de Sobol calculés pour chaque score. Ensuite, à partir de celles-ci, les indices de sensibilité généralisés (GSI) sont calculés (voir Propriété 4.1).

4.3.1 Analyse spatiale avec les composantes principales

Une ACPF sur une base B-splines orthonormalisée a été appliquée sur l'échantillon d'apprentissage (voir la section 3.3.1). On choisit $K = \tilde{K} = 1225$ et $n_{PC} = 5$ (voir la section 3.3.2). Les cinq premières composantes principales (vecteurs propres de taille $\tilde{K}$) ont été utilisées pour construire $\mathrm{GP}^{\mathrm{FPCA}}_{\text{B-splines}}$, notées $\omega_1, \ldots, \omega_{n_{PC}}$. Par conséquent, les

fonctions propres associées de l'ACPF sont analysées. Elles sont aussi appelées composantes principales et assimilées à des vecteurs propres de taille K. Elles sont construites à partir des coefficients des vecteurs propres de la matrice de covariance des $\tilde{K}$ coefficients sélectionnés de la base de fonction $\mathbf{\Phi}(\mathbf{z}) = (\phi_1(\mathbf{z}), \ldots, \phi_K(\mathbf{z}))^\top$ (ici, la base B-splines orthonormalisée). Elles sont notées $\xi_1(\mathbf{z}), \ldots, \xi_{n_{PC}}(\mathbf{z})$ et calculées comme suit :

$$\xi_l(\mathbf{z}) = \sum_{k=1}^{\tilde{K}} \omega_{l,k}\phi_{(k)}(\mathbf{z}), \text{ avec } l = 1, \ldots, n_{PC} \tag{4.6}$$

$\omega_{l,k}$ est le $k^{\text{ème}}$ coefficient du vecteur propre ω_l, avec $k = 1, \ldots, \tilde{K}$ et $l = 1, \ldots, n_{PC}$. $\phi_{(k)}(\mathbf{z})$ est la fonction de la base associée au $k^{\text{ème}}$ coefficient le plus important selon le critère de l'énergie (voir les équations 3.3 et 3.4). Dans la section 3.3, on a $K = \tilde{K}$, donc ici :

$$\xi_l(\mathbf{z}) = \sum_{k=1}^{K} \omega_{l,k}\phi_k(\mathbf{z}), \text{ avec } l = 1, \ldots, n_{PC} \tag{4.7}$$

Les figures 4.1 et 4.2 montrent les cinq premières composantes principales (fonctions propres), et les indices de Sobol des scores correspondant. Les indices du premier ordre et totaux ont été calculés.

Pour analyser chaque composante principale, [Campbell et al., 2006] tracent sur le même graphique la fonction moyenne et les deux fonctions obtenues en ajoutant et soustrayant la fonction propre. Cependant, représenter ou analyser ces trois fonctions, est difficilement faisable dans le cas des données spatiales. En effet, elles peuvent être visualisées seulement avec un graphique à trois dimensions ou avec des courbes de niveaux. Par conséquent, dans les figures 4.1 et 4.2, on ne trace que la fonction propre. Pour l'interprétation, le niveau 0 des courbes de niveaux est assimilé à la fonction moyenne (ici, la moyenne empirique des $n = 200$ cartes de l'échantillon d'apprentissage). Les couleurs chaudes représentent une croissance spatiale par rapport à la moyenne. À l'inverse, les couleurs froides représentent une décroissance spatiale par rapport à la moyenne.

Pour faire l'analyse de sensibilité, il nous faut observer trois informations : la structure des composantes principales (ou fonctions propres), le pourcentage de variance expliquée et les indices de Sobol des scores. On fait les interprétations suivantes :

— La première composante, dont le pourcentage de variance expliquée est 79.2%, représente une décroissance moyenne du coin inférieur gauche au coin supérieur droit. Les variables les plus influentes sont X_6 et X_8, dont les indices de Sobol sont supérieurs à 0.2. X_8 a plus d'influence de par sa variation seule que X_6 : l'indice total est presque égal à l'indice du premier ordre. Pour X_6, l'indice du premier ordre est légèrement inférieur à l'indice total. Ensuite les deux autres variables

influentes sont X_2 et X_4, qui ont principalement aussi une influence seule. Les indices totaux de X_1, X_3, X_5 et X_7 ont une légère influence avec des indices totaux de 0.05, 0.1, 0.05 et 0.05. X_3, X_5 et X_7 ont des indices du premier ordre inférieurs à l'indice total. De plus, leurs valeurs sont proches de 0. Leurs influences proviennent de leurs interactions avec d'autres variables d'entrée.

— La seconde composante principale est séparée en deux parties : une, en dessous de la ligne diagonale verte (représentant le niveau 0), qui représente une croissance moyenne, et une deuxième, au dessus, qui représente une décroissance moyenne jusqu'au coin supérieur droit. La variable la plus influente est X_7, avec un indice du premier ordre entre 0.4 et 0.5, et un indice total à 0.6. La seconde variable influente est X_6, avec un indice total à 0.3. Mais cette influence intervient en interaction avec d'autres variables d'entrées, car l'indice du premier ordre est égal à 0. Il en est de même pour X_3 et X_5 pour lesquels les indices de Sobol totaux sont 0.2 et 0.1. Malgré que les indices de Sobol de X_7 soient bien supérieurs à ceux observés pour la première composante principale, le pourcentage de variance expliquée de la seconde composante est 12.6%, qui est donc environ 6 fois moins importante que celui de la première.

— Le pourcentage de variance expliquée de la troisième composante principale n'est que de 3.2%. Néanmoins, les indices de Sobol calculés permettent de voir que X_3 et X_6 ont une influence forte (indices totaux à 0.7 et 0.6) sur la diagonale du coin supérieur gauche au coin inférieur droit. X_3 et X_6 ont un indice du premier ordre proche de 0, donc leurs influences proviennent de leurs interactions. En effet, mis à par X_3 et X_6, les indices totaux des autres variables sont proches de 0 ou égalent à l'indice du premier ordre.

— Les pourcentages de variance expliquée de la quatrième et cinquième composantes principales sont respectivement 1.6% et 1.4%, qui sont négligeables par rapport aux autres composantes.

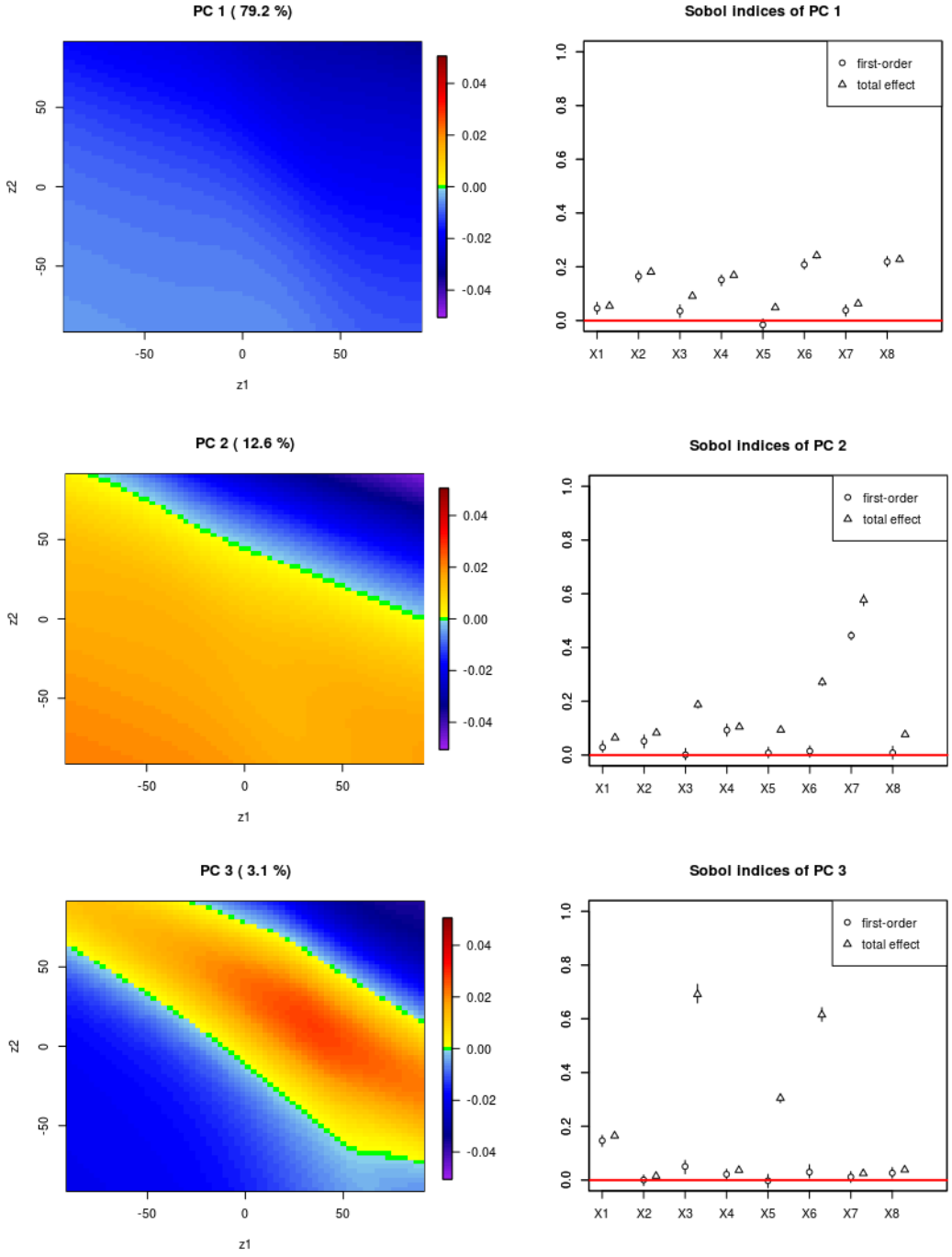

FIGURE 4.1 – La colonne de droite correspond aux trois premières composantes principales de l'ACPF sur une base B-splines orthonormalisée, qui a été appliquée à l'échantillon d'apprentissage de taille $n = 200$ (voir la section 3.3.1). La colonne de gauche correspond aux indices de Sobol calculés pour le score associé. Les indices du premier ordre sont représentés par des cercles. Les indices totaux sont eux représentés par des triangles.

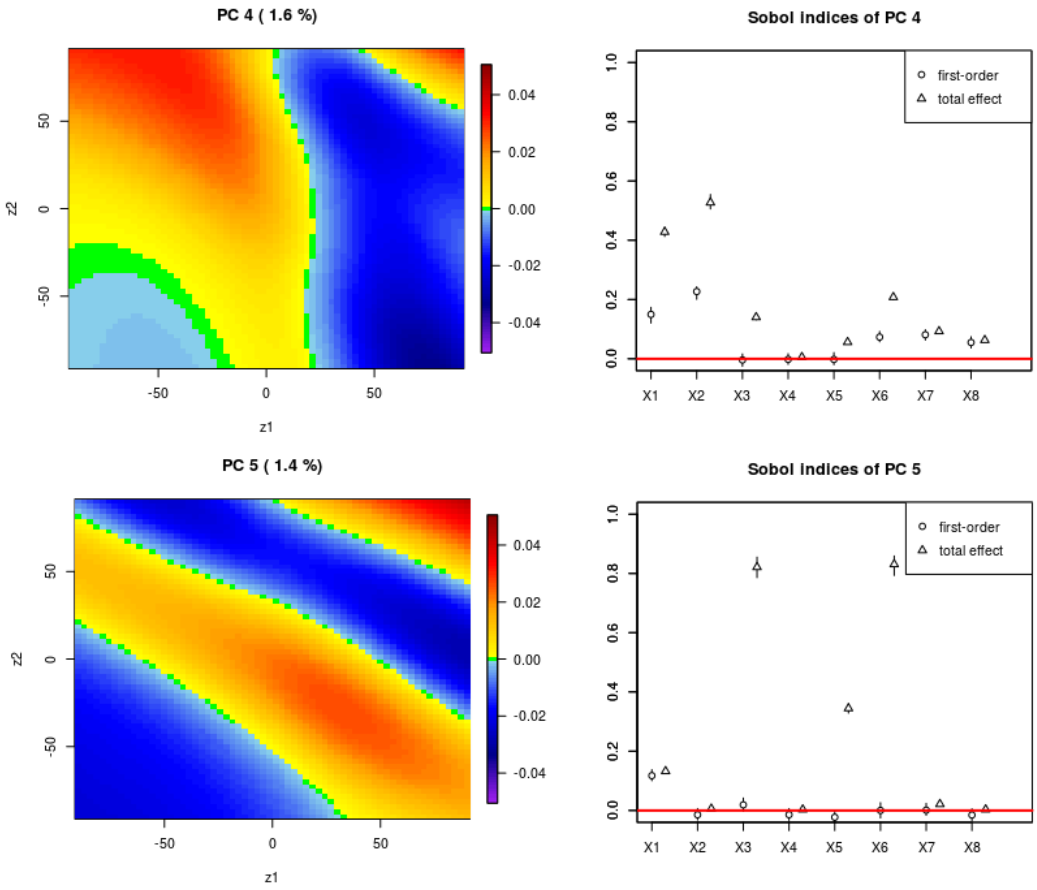

Figure 4.2 – La colonne de droite correspond à la 4−ème et la 5−ème composantes principales de l'ACPF sur une base B-splines orthonormalisée, qui a été appliquée à l'échantillon d'apprentissage de taille $n = 200$ (voir la section 3.3.1). La colonne de gauche correspond aux indices de Sobol calculés pour le score associé. Les indices du premier ordre sont représentés par des cercles. Les indices totaux sont eux représentés par des triangles.

Analyser des cartes d'indices de Sobol, nous aurait emmené à en construire au moins deux par entrée : une, pour les indices du premier ordre, et une, pour les indices totaux. Cela revient à analyser 16 cartes (sans compter les indices du second ordre etc.). L'approche avec l'ACPF a ici limité l'analyse de sensibilité spatiale à 5 cartes. Une analyse de sensibilité spatiale permet d'étudier l'influence $\mathbf{X}$ sur la structure spatiale de $y_{\mathbf{x}}(.)$. De plus, dans des applications réelles, les fonctions propres peuvent avoir un sens physique. Par exemple, pour les submersions marines, elles permettent de voir les variables influençant des inondations plus fortes dans des zones spécifiques. Néanmoins, il est fastidieux de faire une hiérarchie bien définie des variables d'entrée. C'est pourquoi, les indices GSI (voir Définition 4.1) sont aussi calculés, car ils facilitent la hiérarchisation des variables, en concentrant l'information sur un seul critère.

4.3.2 Analyse de sensibilité avec l'indice général

La propriété 4.1 indique que GSI est équivalent à la moyenne des indices de Sobol calculés sur les composantes principales, pondérée par les valeurs propres correspondantes. Les indices de Sobol ont été calculés dans la section précédente (4.3.1). La figure 4.3 montre les estimations des indices de sensibilité généralisés (GSI). X_6 est la variable la plus influente avec l'indice total le plus élevé. X_8 est la seconde variable la plus influente, dont son effet principal (c'est-à-dire son influence seule) est égal à celui de X_6. On remarque que son influence est entièrement définie par son effet principal (l'indice du premier ordre est égal à l'indice total). X_2, X_4 et X_7 sont eux aussi entièrement définis par leurs influences seules. Elles correspondent aux 3^e, 4^e, et 5^e variables les plus influentes. X_1, X_3, et X_5 sont les trois variables les moins influentes. X_1 est entièrement défini par son effet principal. Finalement, X_3 et X_5 ont principalement de l'influence en interaction avec les autres variables d'entrée (indice du premier ordre proche de 0).

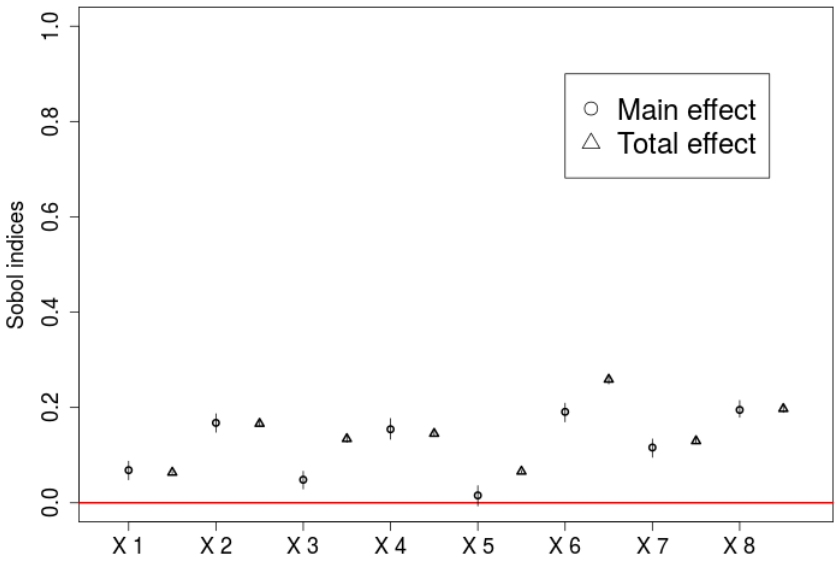

FIGURE 4.3 – Estimations des indices de sensibilité spatiale généralisé (GSI) des 8 variables d'entrées. Les indices du premier ordre sont représentés par des cercles. Les indices totaux sont eux représentés par des triangles.

Troisième partie

Application à la submersion marine

Chapitre 5

Application aux modèles d'aléa du BRGM et de la CCR

Sommaire

La méthodologie décrite dans la section 3.2.1.1 est appliquée au cas des submersions marines. Le site d'étude est le village des Bouchôleurs, situé sur la commune d'Yves (17 340) : côte Atlantique française, près de « La Rochelle », qui a été touché par la tempête Xynthia en 2010 (voir Figure 5.1). Le principal processus de submersion correspond à du débordement. On nomme modèle aléa, les modèles simulant le processus de submersion marine, et permettant d'estimer les emprises submergées et les hauteurs d'eau associées. Dans ce chapitre, on étudie deux modèles d'aléa : celui du BRGM et celui de la CCR. Les simulations du BRGM sont obtenues avec le code numérique en différence finie MARS [Lazure and Dumas, 2008], où des adaptations ont été faites par le BRGM pour prendre en compte les spécificités des processus d'inondation locaux (processus hydrauliques autour des connexions comme les buses, déversoirs, etc. et les phénomènes de brèche) [Rohmer et al., 2018]. Le modèle de la CCR est également un modèle de type différences finies, basé sur le modèle Lisflood-FP [Bates et al., 2005] [Bates et al., 2010].

5.1 Présentation de l'analyse de sensibilité

La première analyse de sensibilité (voir la section 5.4) se concentre sur l'influence de la marée et de la surcote sur la distribution spatiale de la profondeur maximale de l'eau après une inondation. Ici, l'évolution temporelle des deux signaux est simplifiée : la marée est assimilée à une courbe sinusoïdale d'amplitude T (entre $0,95m$ et $3,70m$) ; la surcote est supposée triangulaire (voir figure 5.1) en utilisant quatre paramètres, à savoir S le pic de surcote (compris entre 0,65m et 2m), t_0 la différence de phase entre le pic de surcote et le pic de la marée (entre -6 et 6 heures), t_+ et t_- la durée de l'augmentation et de la diminution de la surcote (entre 0,5 et 12 heures). On s'intéresse à la sensibilité des cinq paramètres d'entrée $\mathbf{x} = (T, S, t_0, t_+, t_-)$. Les entrées sont récapitulées dans le Tableau 5.1 et illustrées par (b) dans la Figure 5.1.

Entrée	**Notation**	**Intervalle de valeurs**
Amplitude de la marée	T	[0.95m ,3.70m]
Pic de surcote	S	[0.65m ,2m]
Différence de phase entre le pic de surcote et l'amplitude de la marée	t_0	[−6h ,6h]
Durée de l'augmentation de la surcote	t_+	[0.5h , 12h]
Durée de la diminution de la surcote	t_-	[0.5h ,12h]

Tableau 5.1 – Entrées des modèles aléa.

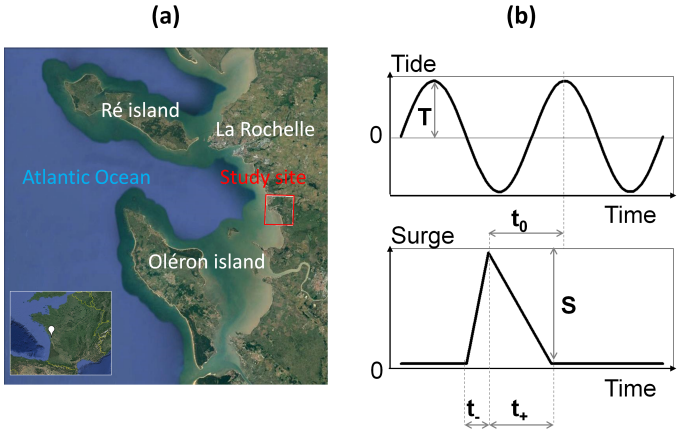

Figure 5.1 – a) Localisation du site, b) Paramétrage de l'évolution temporelle de la marée (en anglais, « Tide ») et de la surcote (en anglais, « Surge »).

La deuxième analyse de sensibilité (voir la section 5.5) porte sur d'autres paramètres du modèle d'aléa : le forçage marin, les connexions hydrauliques, et les coefficients de rugosité. On étudie leurs influences sur l'estimation de l'inondation, lors de la tempête Xynthia en 2010, à La Rochelle. Les détails sont donnés dans la section 5.5.

La sortie du simulateur est une carte de profondeur d'eau maximale atteinte durant l'évènement. La hauteur d'eau est notée HE. Elle correspond à une carte avec des discrétisations régulières, de 194×202 pour l'aléa du BRGM, et de 213×213, pour l'aléa de la CCR. Les pixels des deux cartes de sortie ont une résolution spatiale de $25m \times 25m$. Cependant, elles n'ont pas la même projection géographique [Kennedy et al., 2000]. Afin de comparer les deux modèles, les cartes résultant des simulations ont été mises dans la même projection en Lambert93. De plus, pour que la transformation en ondelettes soit possible, elles sont aussi projetées sur une grille de dimension dyadique : 256×256.

La Figure 5.2 donne des exemples de cartes de profondeurs d'eau estimées par les deux modèles. En fonction des valeurs des entrées, l'étendue spatiale de l'inondation est plus ou moins importante pour les deux modèles. Sur le fond de carte, on remarque tout d'abord des infrastructures structurelles contraignant l'inondation : la route principale locale (ligne grise) et la route nationale (ligne rouge). Les deux routes construites légèrement plus haut que les alentours (sur les talus) pour éviter les inondations routières, limitent la pénétration de l'eau à l'intérieur des terres. Cependant, elles n'agissent pas complètement comme une digue (existence de liaisons hydrauliques entre les zones est et ouest de la route).

Pour les cartes du BRGM, on remarque des irrégularités dans la zone correspondant à la zone bleu clair de la carte de droite. Cette zone correspond à la localisation de la principale zone urbaine sur le site d'étude. Pour les cartes de la CCR, les plus hautes profondeurs d'eau sont situées dans la zone noire, qui correspond à un marais. Dans ces valeurs, la profondeur du marais est considérée, ce qui ne traduit donc pas forcément une plus forte inondation dans cette zone. Pour les deux aléas, on remarque aussi que le niveau d'eau peut fortement varier d'un pixel à l'autre. Ces changements brusques s'expliquent par les différences d'altitudes sur le MNT. La figure 5.2 illustre la complexité et l'hétérogénéité des cartes d'inondations.

Pour une seule simulation, les temps de calcul des simulateurs sont d'environ cinq minutes pour la CCR, et entre trente minute et une heure pour le BRGM. À cause de ce temps de calcul, seul un nombre limité de simulations est réalisé en choisissant aléatoirement les configurations de $\mathbf{x}$ selon une séquence aléatoire de Sobol [Bratley and Fox, 1988]. Le même plan d'expériences est utilisé pour les deux modèles d'aléa. Ce type de plan permet de couvrir tout l'espace des configurations uniformément.

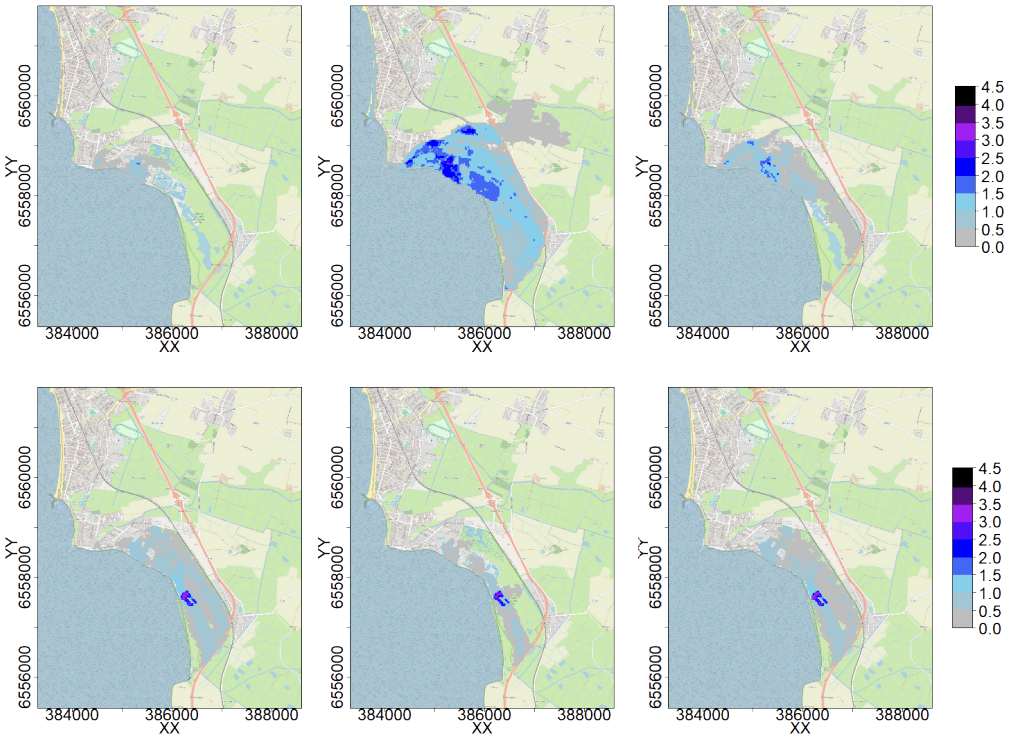

Figure 5.2 – Les trois cartes du haut correspondent à des exemples de cartes de profondeurs d'eau (en m) simulées par le BRGM. Celles du bas sont simulées par le modèle d'aléa de la CCR. De gauche à droite, les entrées du modèle sont respectivement $\mathbf{x_1}$ =(3.61 m, 1.75 m, 5.72 h, -3.10 h, 2.11 h), $\mathbf{x_2}$ =(3.51 m, 1.68 m, 3.93 h, -5.82 h, 5.85 h), et $\mathbf{x_3}$ =(3.23 m, 1.55 m, 0.19 h, -3.66 h, 3.06 h). La couche de fond (SCAN 25® de l'Institut National d'Information Géographique et Forestière IGN) indique les localisations des zones urbaines et les éléments topographiques clés (routes, voies ferrées, marais, etc.).

5.2 Analyse des données

Une méthode simple pour faire l'analyse de sensibilité (AS), est de tracer la sortie du modèle en fonction de chaque paramètre d'entrée - méthode d'analyse des nuages de points (scatter-plot) [Iooss, 2011][Iooss and Lemaître, 2015]. Cependant, ici, la sortie est une matrice de dimension 256×256 avec les HE estimés. Tracer les $n = 500$ cartes, en fonction de chaque entrée, est difficilement réalisable. D'autres possibilités pour représenter les cartes sont de tracer les boxplots et les fonctions de densité de chacune d'elle, comme par exemple dans la Figure 5.3. Avec ces figures, une comparaison entre les cartes est possible. Par exemple, on peut comparer l'intervalle et les quantiles des HE. Mais il est difficile de faire une interprétation pour l'AS.

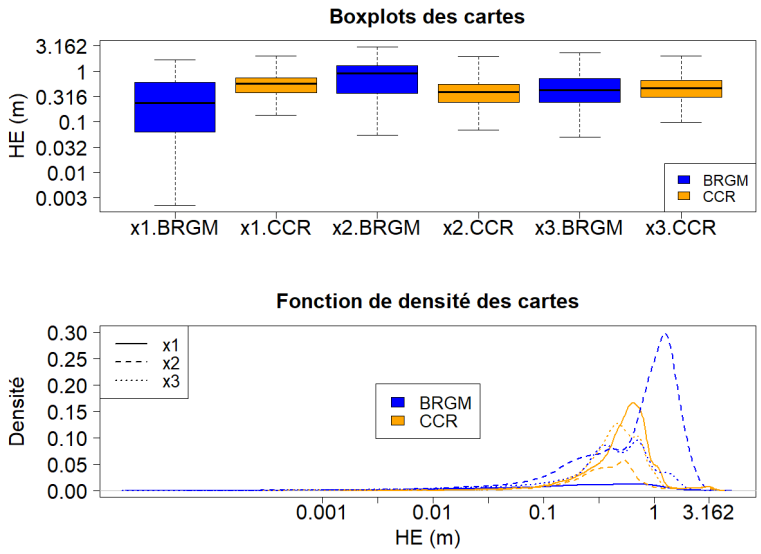

FIGURE 5.3 – Exemples de boxplots et de fonctions de densité (de haut en bas) de cartes de profondeurs d'eau. Les HE sont tracés en échelle $\log_{10}$. Les cartes associées sont celles de la Figure 5.2. Les entrées du modèle sont $\mathbf{x_1}$ =(3.61 m, 1.75 m, 5.72 h, -3.10 h, 2.11 h), $\mathbf{x_2}$ =(3.51 m, 1.68 m, 3.93 h, -5.82 h, 5.85 h), et $\mathbf{x_3}$ =(3.23 m, 1.55 m, 0.19 h, -3.66 h, 3.06 h).

Une solution envisageable est de tracer la moyenne spatiale des HE en fonction de chaque entrée. Pour chaque carte i, $i \in \{1, \ldots, n\}$, la moyenne est calculée par (5.1).

$$\mu_{HE}^{(i)} = \frac{1}{D_{\mathbf{z}}} \sum_{j=1}^{D_z} y_i(\mathbf{z}_j), \ i = 1, \ldots, n \tag{5.1}$$

avec $D_{\mathbf{z}} = 256 \times 256 = 65\ 536$ la dimension des cartes, et $y_i(.)$ la $i$ème carte de l'échantillon. Le tracé de (5.1) en fonction de T, S, t_0, t_-, et t_+ est fait dans la Figure

5.4. Pour commencer, une moyenne à zéro correspond à une carte où aucune inondation est estimée. On constate qu'une partie des simulations des deux modèles correspond à des moyennes à zéro. En effet, les conditions au large sont insuffisantes pour générer du débordement. La Figure 5.5 illustre le nombre de simulations pour lesquelles aucune inondation est estimée, car les conditions au large sont insuffisantes pour générer du débordement. Le BRGM et la CCR ont respectivement 282 et 257 cartes où aucune inondation est estimée, dont 77% et 84% sont communes entre les deux modèles.

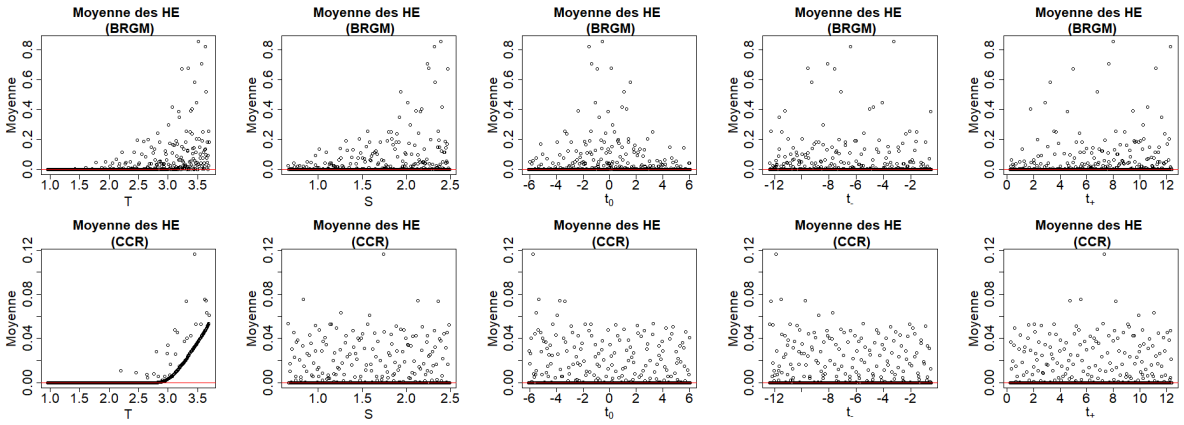

Figure 5.4 – Moyenne des profondeurs d'eau en fonction de chaque paramètre d'entrée de l'aléa du BRGM et de la CCR (de haut en bas), les entrées étant T, S, t_0, t_-, et t_+ (de gauche à droite). En noir, le nuage de points des moyennes des HE. En rouge, le trait situant la moyenne à zéro.

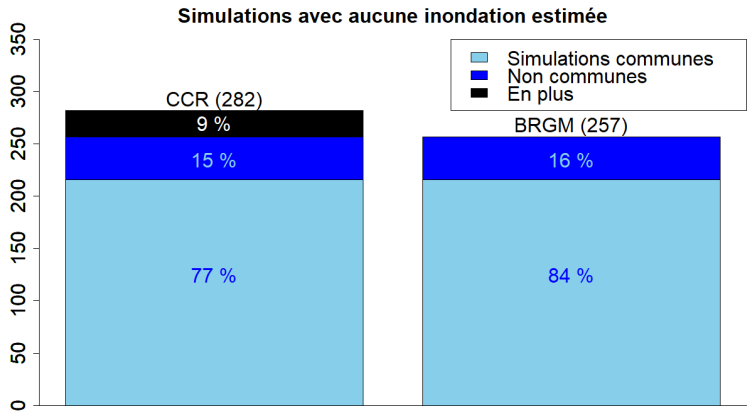

Figure 5.5 – Barplot des simulations pour lesquelles aucune inondation est estimée.

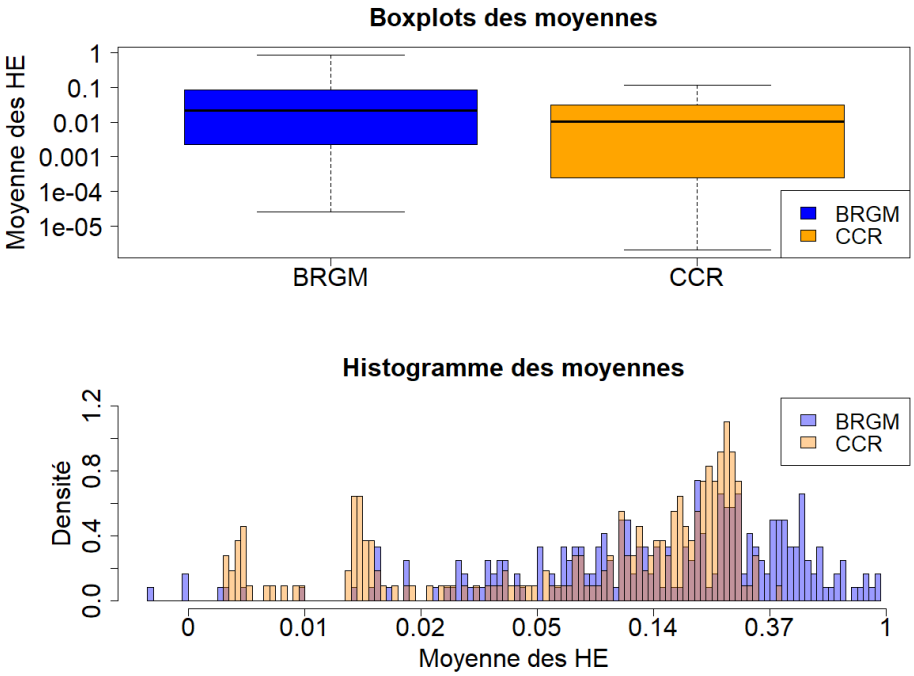

FIGURE 5.6 – Boxplots et histogramme de la moyenne des HE, en orange pour l'aléa de la CCR, et en bleu pour l'aléa du BRGM. Les moyennes sont tracées en échelle $\log_{10}$. Les moyennes nulles (μ_{HE}=0) ne sont pas prises en compte dans les figures.

Pour l'aléa de la CCR, on remarque une forte influence de T (voir la figure 5.4). On y observe un effet de seuil : de $0.95m$ à $3m$, le modèle n'estime aucune inondation, à partir de $3m$, on voit une tendance linéaire et croissante de la moyenne des HE. Les autres paramètres pris indépendamment ne semblent avoir aucune influence. Pour l'aléa du BRGM, on remarque aussi une forte influence de T, mais aussi de S et t_0. Pour t_0, le nuage de points semble suggérer une croissance de la moyenne de $-6h$ à $0h$, puis une décroissance jusqu'à $6h$.

On représente l'ensemble des $(\mu_{HE}^{(i)}) > 0$, avec $i = 1, \ldots, n$ (voir (5.1)) sous la forme de boxplots et d'histogrammes (voir la figure 5.6). Les moyennes nulles (μ_{HE}=0) ne sont pas prises en compte dans les figures. En effet, on cherche ici à visualiser la différence de l'estimation entre les deux modèles lorsqu'une inondation se produit. De plus, les moyennes sont tracées en échelle $\log_{10}$. La médiane et le 3ème quartile des $(\mu_{HE}^{(i)})$ de la CCR est inférieure ou égale à la médiane des $(\mu_{HE}^{(i)})$ du BRGM. Sur l'histogramme, on observe des densités plus fortes et des moyennes plus faibles pour la CCR. Pour les moyennes les plus fortes, la densité de la CCR est nulle là où celle du BRGM ne l'est pas. Cela illustre que les inondations estimées par l'aléa de la CCR sont globalement moins étendues et moins importantes que celles simulées par le BRGM.

La Figure 5.7 représente les cartes moyenne et écart-type des estimations des aléa. Elles sont respectivement calculées suivant (5.2) et (5.3).

$$\mu_y(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} y_i(\mathbf{z}) \tag{5.2}$$

et

$$\sigma_y(\mathbf{z}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i(\mathbf{z}) - \mu_y(\mathbf{z}))^2} \tag{5.3}$$

À partir de la figure 5.7, concernant l'aléa du BRGM, on observe qu'en moyenne les plus grandes HE sont situées au niveau de la zone urbaine (zone violette). L'écart-type est aussi le plus grand dans cette zone. Concernant l'aléa de la CCR, l'inondation est en moyenne moins étendue que les estimations du BRGM. Les plus hauts HE sont situés dans la zone où se situe un marais (zone bleu foncé). L'écart-type y est aussi le plus grand.

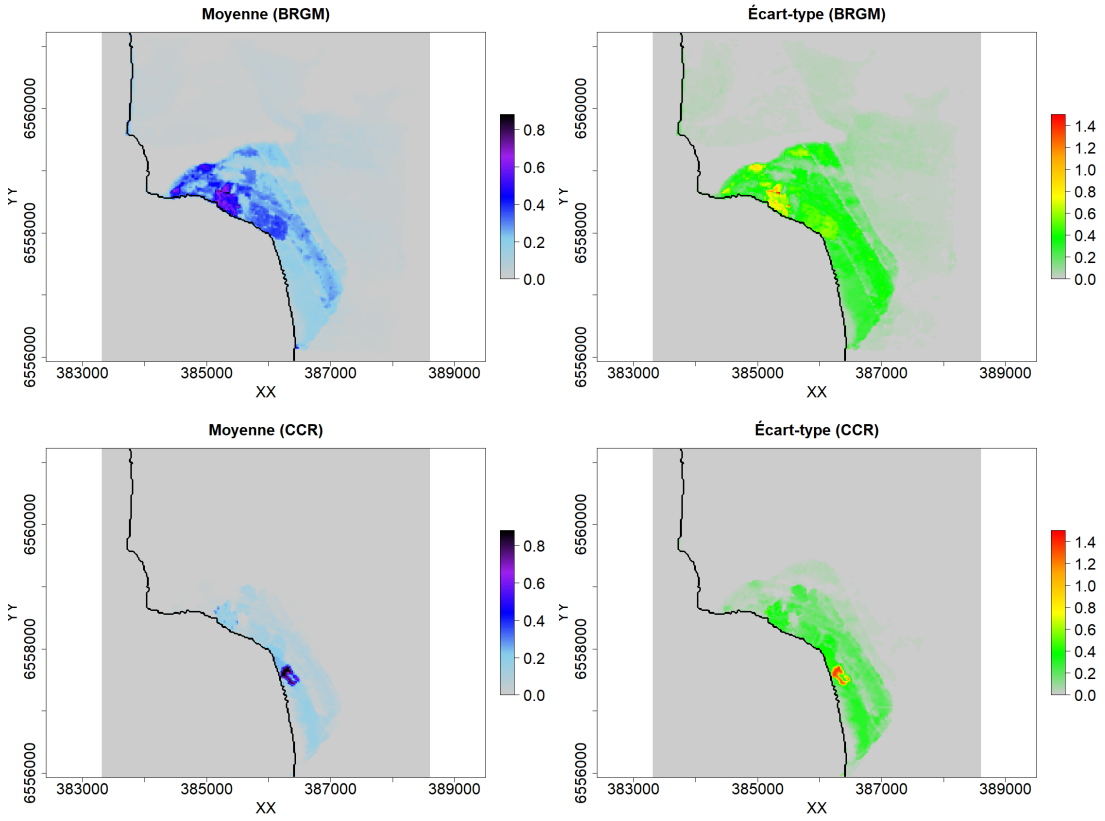

Figure 5.7 – De gauche à droite, les cartes moyenne et écart-type des estimations.

5.3 Méta-modélisation des modèles d'aléa

Les trois méthodes de méta-modélisation GP^{PCA}, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, et $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, présentées dans la section 3.3, sont maintenant comparées dans le cas des submersions marines. Les méta-modèles sont entraînés en utilisant un échantillon d'apprentissage de taille $n_{learning} = 400$. Ces observations sont aléatoirement choisies parmi les 500 simulations des deux modèles. Pour cela, les mêmes simulations utilisées dans [Perrin et al., 2021] ont été reprises. Partant de l'échantillon simulé par le BRGM, 200 observations sont choisies aléatoirement dans l'échantillon des cartes non inondées, puis les 200 autres sont prises parmi les cartes inondées. Les échantillons d'apprentissage des deux modèles sont constitués des mêmes simulations. Afin de tester la précision de prédiction des méta-modèles, les $n_{test} = 100$ observations restantes sont utilisées comme échantillon test.

5.3.1 ACP fonctionnelle (ACPF)

5.3.1.1 Définition des bases de fonctions

Pour $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, les ondelettes de Daubechies D4 sont utilisées, afin d'avoir un bon compromis entre la taille du support et de la sélectivité dans le domaine fréquentiel. On fixe la profondeur de la décomposition à $J = 1$, afin de représenter les hétérogénéités présentes sur les cartes, notamment dans les zones urbaines.

Pour $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, des splines de degré 1 sont utilisées dans le même but. La dimension de la base est définie par le nombre de nœuds pris dans le domaine spatiale. Les coordonnées spatiales sont notées XX et YY. Le même nombre de nœuds (ou knots, en anglais) est considéré pour chacune d'elles. Pour calibrer la taille de la base B-splines pour chaque aléa, les cartes inondées des échantillons d'apprentissage sont approximées dans des bases B-splines pour des nombres de nœuds différents. La racine de l'erreur quadratique moyenne entre les cartes de l'échantillon d'apprentissage et leurs approximations dans la base est calculée comme suit :

$$RMSE_{splines}(z) = \sqrt{\frac{1}{n_{learning}} \sum_{i=1}^{n_{learning}} (y_{\mathbf{x}_i}(\mathbf{z}) - \hat{y}^{splines}_{\mathbf{x}_i}(\mathbf{z}))^2} \tag{5.4}$$

avec $y_{\mathbf{x}_i}(.)$, la i^e simulation de l'échantillon d'apprentissage, $i = 1, \ldots, 400$, et $\hat{y}^{splines}_{\mathbf{x}_i}(.)$, son approximation dans la base B-splines. On observe ensuite les boxplots des cartes de RMSE obtenues (voir la Figure 5.8). Le nombre maximum de nœuds testé est 100 pour chaque dimension, soit au total 10 000 nœuds. La dimension des cartes étant $n_{\mathbf{z}} = 256 \times 256 = 65\,536$, la base est représentée par une matrice de dimension $n_{\mathbf{z}} \times K$, avec K la dimension de la base. Pour $K > 10\,000$, $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$devient coûteux en temps de calcul

et en mémoire. La complexité de l'étape d'orthonormalisation est $O(n_{\mathbf{z}}K^2)$, et celle de l'estimation des coefficients et de reconstitution des cartes est $O(n_{\mathbf{z}}Kn_{learning})$.

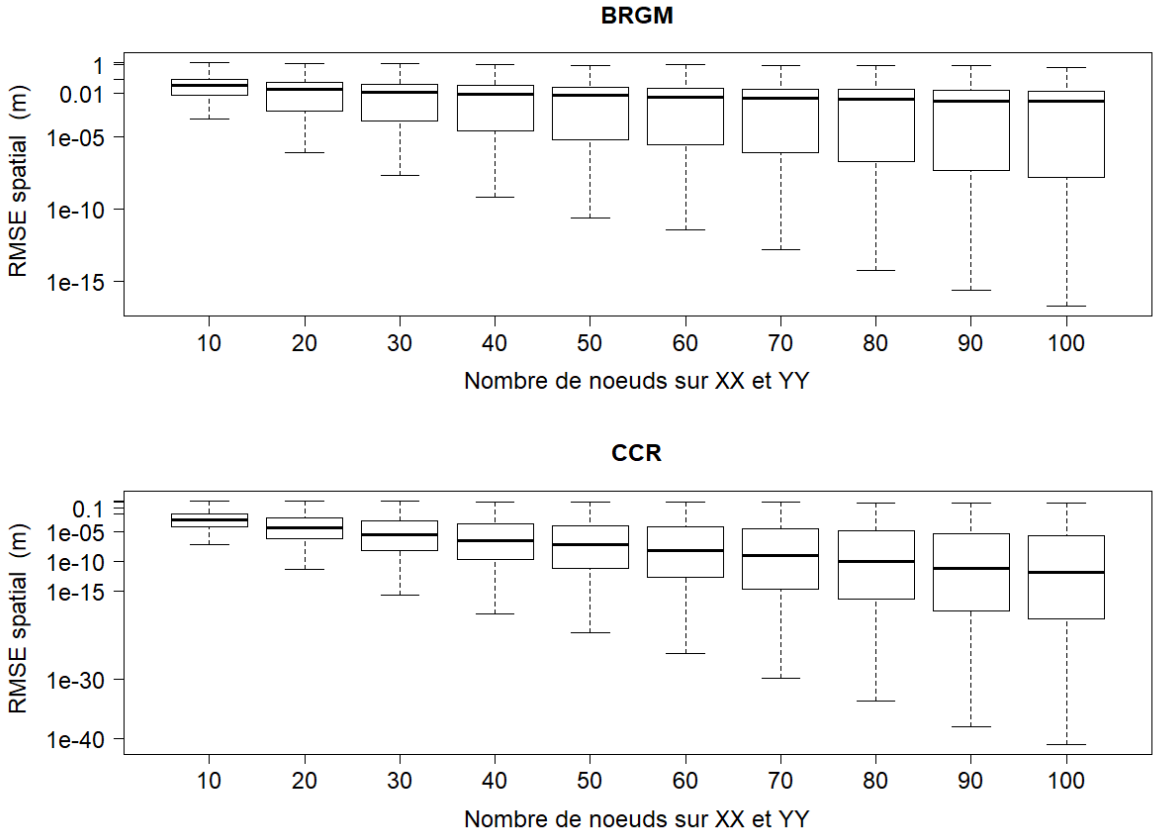

Figure 5.8 – Boxplots des cartes de $RMSE_{splines}(.)$: en haut, ceux pour l'aléa du BRGM, en bas, ceux pour l'aléa de la CCR. Les $RMSE_{splines}(.)$ sont représentés en l'échelle $\log_{10}$. Notez la différence d'échelle des axes des ordonnées entre les deux graphiques.

5.3.1.2 Caractéristiques de l'ACP fonctionnelle

Avant de procéder à la prédiction, on analyse les caractéristiques de l'ACP multivariée standard, de l'ACPF sur base d'ondelettes, et de l'ACPF sur base B-splines orthonormalisée, appliquées aux cartes de l'échantillon d'apprentissage.

Dans un premier temps, le nombre de composantes principales, noté n_{PC}, de la méthode GP$^{\text{PCA}}$est calibré par validation croisée. L'échantillon d'apprentissage de taille $n_{learning} = 400$ est divisé en $k = 10$ sous-échantillons. Les sorties d'un des sous-échantillons sont supposées inconnues et estimées par le méta-modèle construit à partir des observations des $k-1 = 9$ autres sous-échantillons. Le procédé est répété pour chaque sous-échantillon.

L'erreur de prédiction est calculée à chaque itération par la racine de l'erreur quadratique moyenne (5.5). Ensuite, n_{PC} est choisi tel que l'on minimise la moyenne (5.6).

$$RMSE_{CV}^{(l)}(\mathbf{z}) = \sqrt{\frac{1}{n_l}\sum_{j=1}^{n_l}(y_{j,l}(\mathbf{z}) - \hat{y}_{j,l}(\mathbf{z}))^2} \tag{5.5}$$

$$\overline{RMSE_{CV}}(\mathbf{z}) = \frac{1}{k}\sum_{l=1}^{k} RMSE_{CV}^{(l)} \tag{5.6}$$

avec n_l la taille du l^e sous-échantillon de la validation croisée, $l = 1, \ldots, k$, $y_{j,l}(\mathbf{z})$ la j^e carte du sous-échantillon, et $\hat{y}_{j,l}(\mathbf{z})$ son estimation. $\overline{RMSE_{CV}}(\mathbf{z})$ a été calculé pour $n_{PC} \in \{1, 2, \ldots, 10\}$.

Les boxplots des 10 cartes de (5.6) sont comparés dans la figure 5.9. Le 1^{er} quartile, la médiane, et le 3^e quartile, étant égaux à 0 pour la CCR, des quantiles entre 75% et 100% sont aussi tracés. Pour le BRGM, le 3^e quartile minimum est atteint à partir de $n_{PC} = 2$. De plus, la forme des boxplots reste la même à partir de $n_{PC} = 2$. Il en est de même pour le quantile 95% pour le modèle de la CCR. Finalement, on choisit $n_{PC} = 2$ pour les deux modèles d'aléa. Les pourcentages de variance expliquée des composantes principales sont représentés dans la figure 5.10, pour chaque itération de la validation croisée. On remarque que les pourcentages sont similaires quel que soit l'itération.

Les deux premières composantes principales obtenues à partir des deux modèles, sont représentées dans la figure 5.11. Elles sont ensuite comparées à celles obtenues par ACPF sur base d'ondelettes et sur base B-splines orthonormalisée (voir les figures 5.12, 5.13, 5.14, et 5.15). Les configurations suivantes de l'ACPF ont été considérées :

1. Configuration ACPF 1 : L'ACP a été appliquée à tous les coefficients de la base de fonctions.
2. Configuration ACPF 2 : L'ACP a été appliquée au minimum de coefficients telle que soit reproduit la totalité de l'énergie moyenne, c'est-à-dire telle que la somme des $\tilde{K}$ premiers (3.4) soit égale à $p = 1$.
3. Configuration ACPF 3 : L'ACP a été appliquée au minimum de coefficients telle que soit reproduit au maximum 99% de l'énergie moyenne, c'est-à-dire telle que la somme des $\tilde{K}$ premiers (3.4) soit inférieure ou égale à $p = 0.99$.

Afin de faciliter la lecture, ces configurations sont numérotées de 1 à 3. Les tableaux 5.2 et 5.3 indiquent respectivement les temps de calcul des trois méthodes, pour les modèles d'aléa du BRGM et de la CCR : ceux de la décomposition sur les bases de fonctions et ceux de l'ACP. Le nombre de coefficients, auxquels sont appliqués l'ACP, y est aussi renseigné. Les codes ont été exécutés sur un cœur d'un AMD Ryzen$^{\text{TM}}$ 7 4700U CPU, sauf la décomposition sur base B-splines pour laquelle 5 cœurs ont été utilisés.

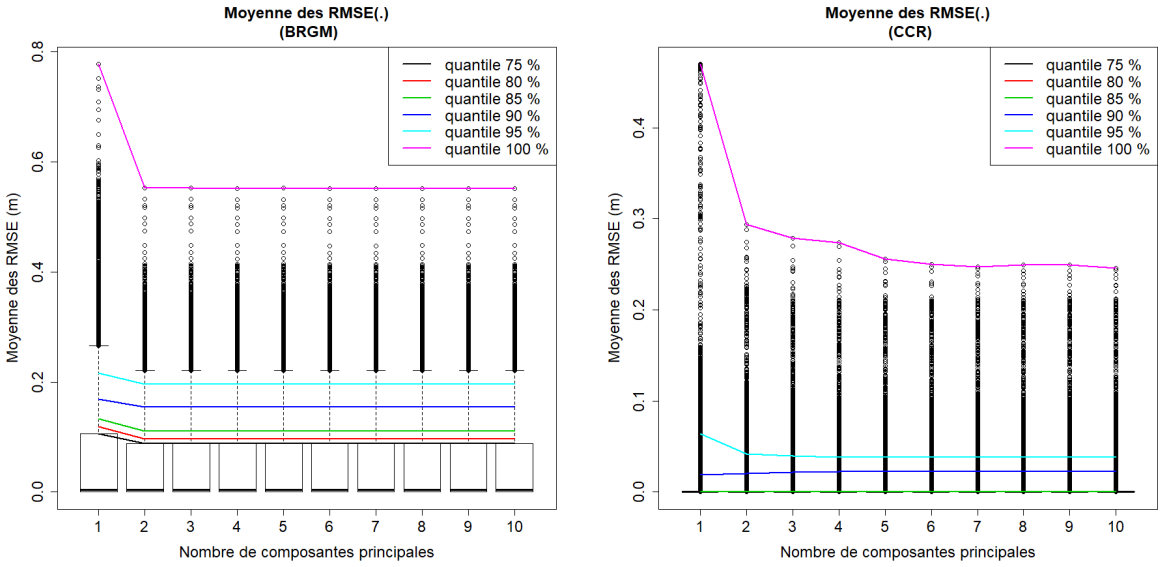

Figure 5.9 – GP$^{\text{PCA}}$: Boxplots du RMSE moyen de la validation croisée. À gauche, pour le modèle d'aléa du BRGM, et à droite, pour celui de la CCR.

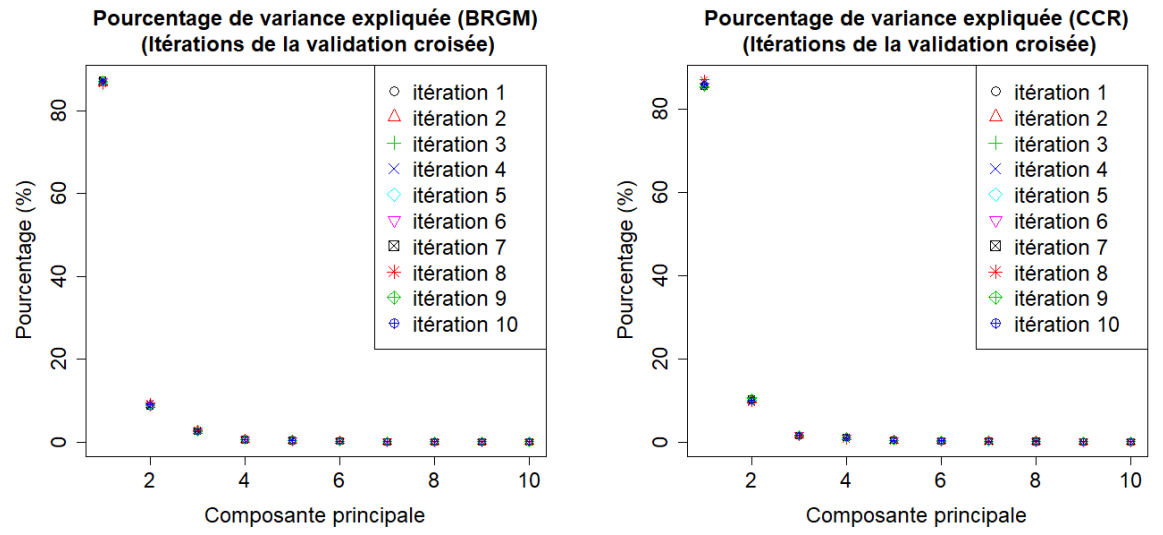

Figure 5.10 – GP$^{\text{PCA}}$: Pourcentage de variance expliquée des composantes principales, pour chaque itération de la validation croisée.

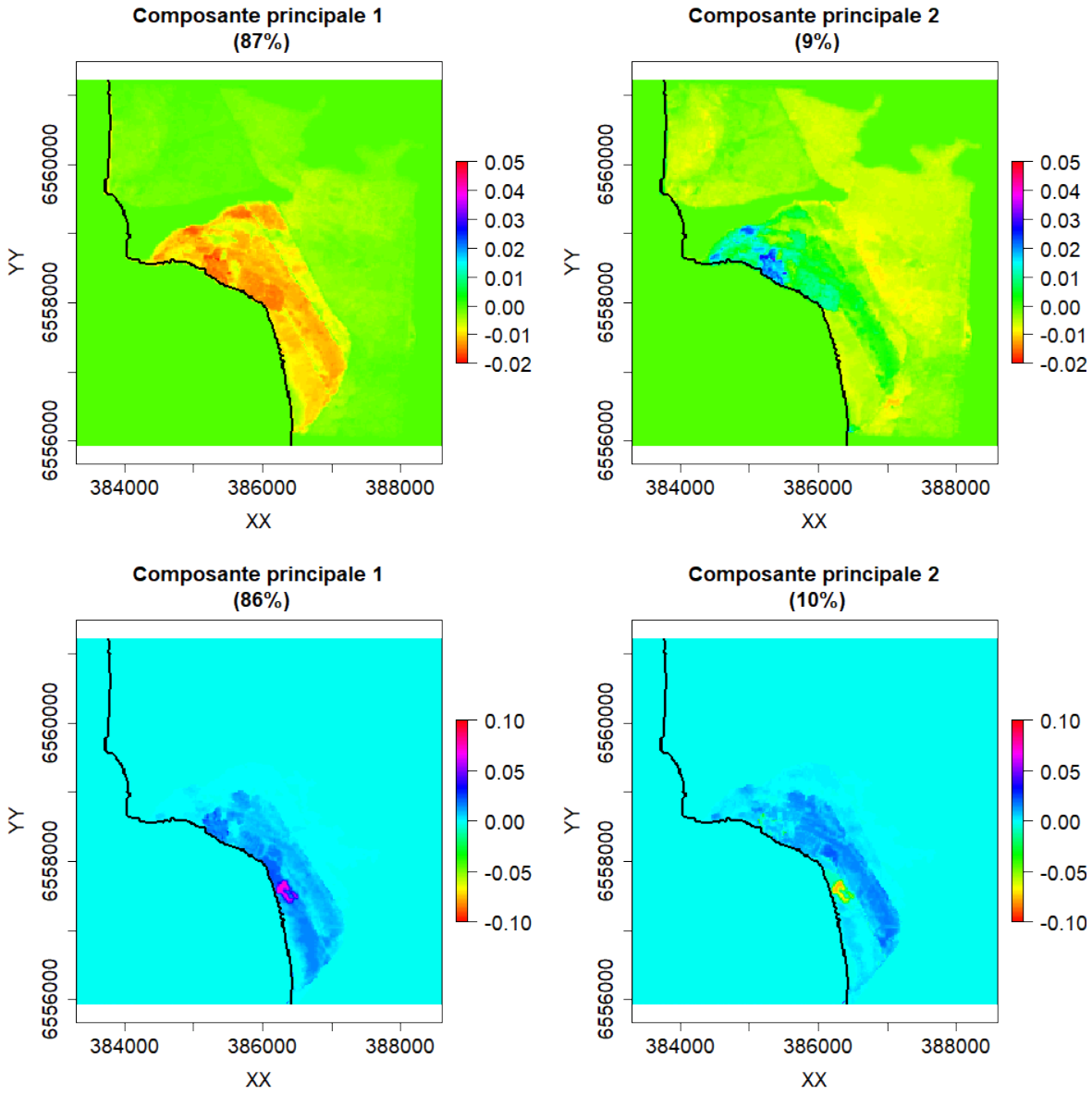

FIGURE 5.11 – GP^{PCA} : Les deux premières composantes principales. En haut, celles obtenues avec le modèle d'aléa du BRGM. En bas, celles obtenues avec le modèle d'aléa de la CCR. Le trait noir correspond au trait de côte.

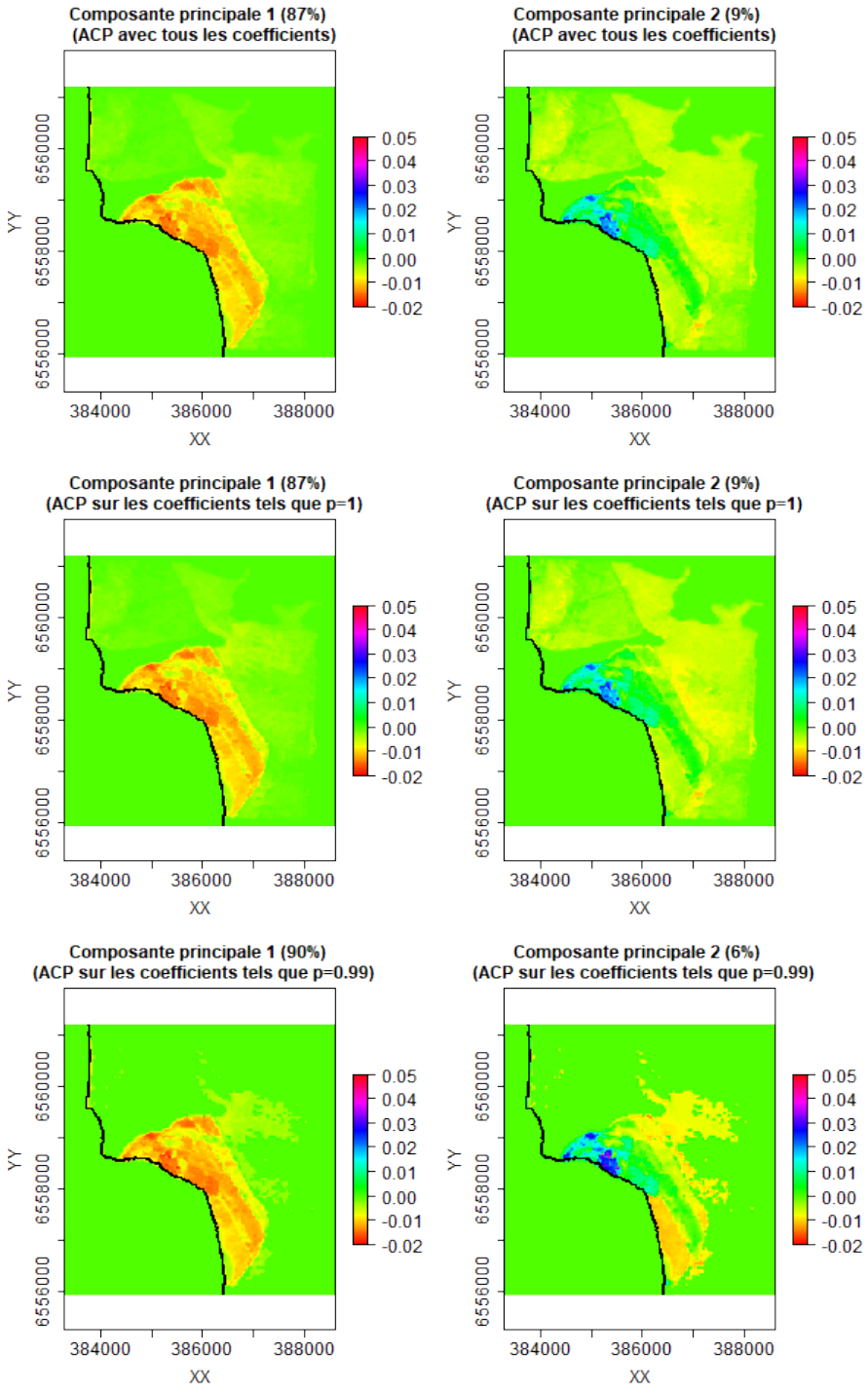

Figure 5.12 – $GP^{FPCA}_{wavelet}$, modèle BRGM : Les deux premières composantes principales. De haut en bas, pour une ACP appliquée à tous les coefficients de la base, et aux coefficients telle que $100p\%$ de l'énergie moyenne soit reconstituée (voir 3.4), avec $p = 1$ et $p = 0.99$. Le trait noir correspond au trait de côte.

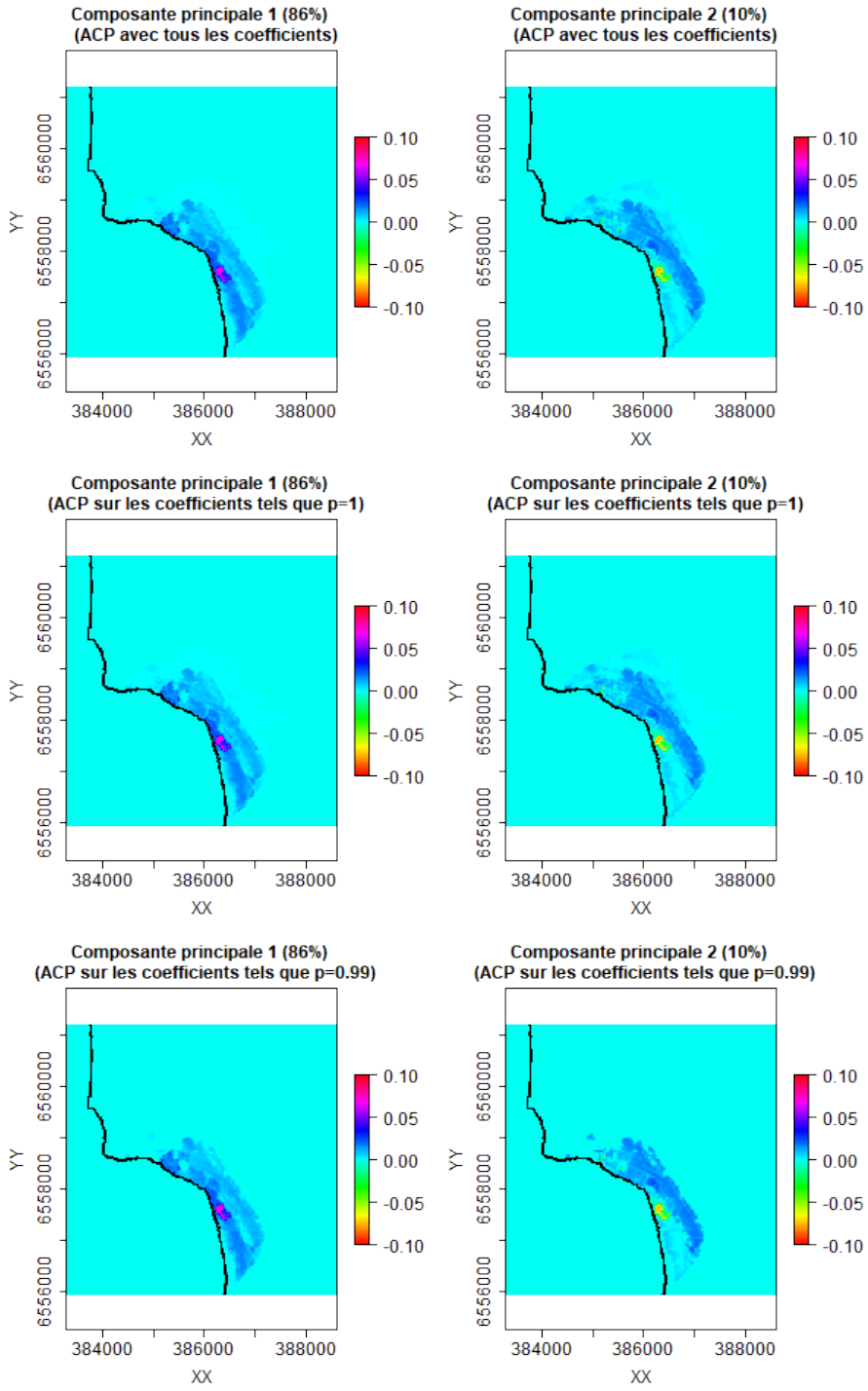

FIGURE 5.13 – $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, modèle CCR : Les deux premières composantes principales. De haut en bas, pour une ACP appliquée à tous les coefficients de la base, et aux coefficients telle que $100p\%$ de l'énergie moyenne soit reconstituée (voir 3.4), avec $p = 1$ et $p = 0.99$. Le trait noir correspond au trait de côte.

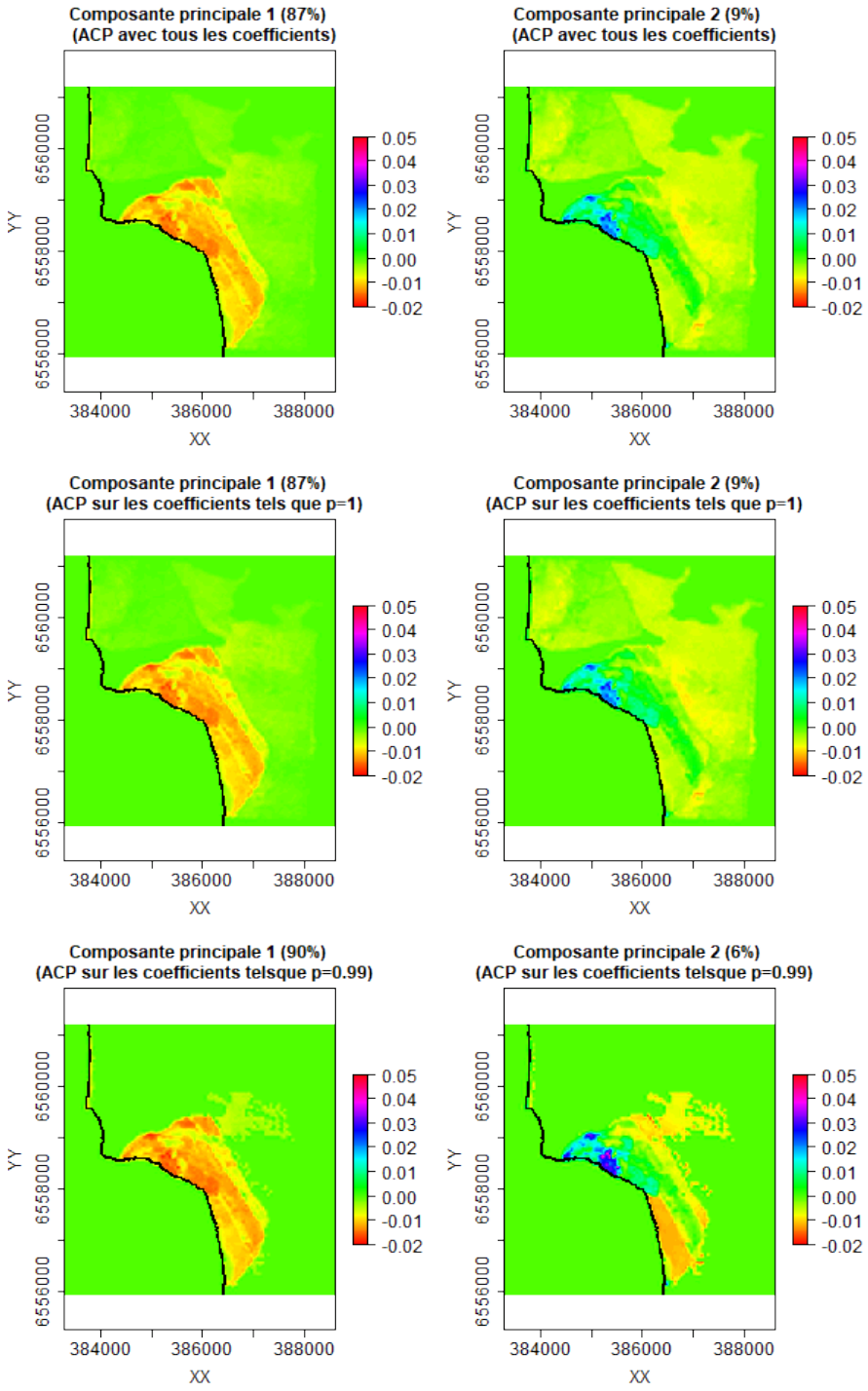

Figure 5.14 – $GP^{FPCA}_{\text{B-splines}}$, modèle BRGM : Les deux premières composantes principales. De haut en bas, pour une ACP appliquée à tous les coefficients de la base, et aux coefficients telle que $100p\%$ de l'énergie moyenne soit reconstituée (voir 3.4), avec $p = 1$ et $p = 0.99$. Le trait noir correspond au trait de côte.

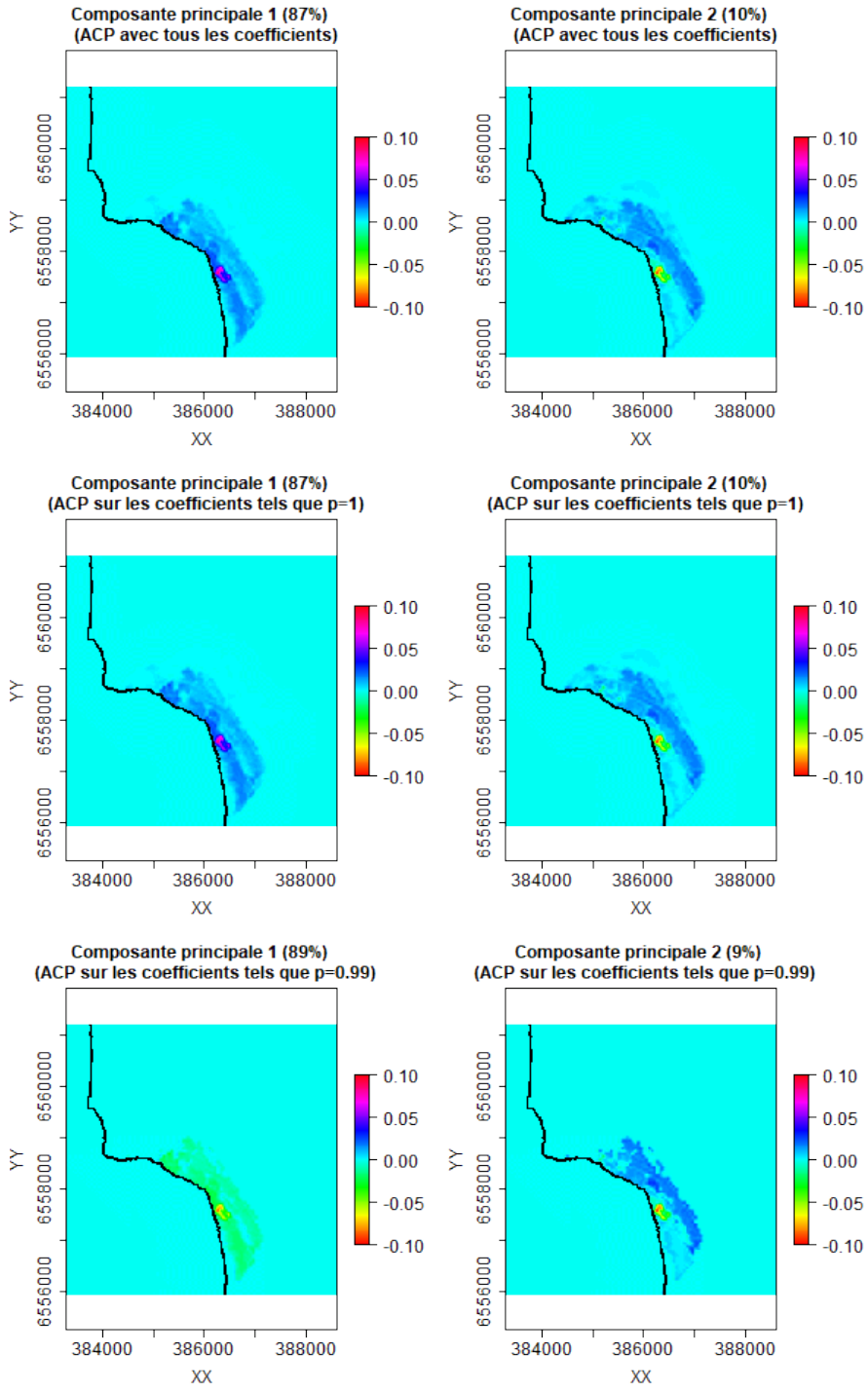

FIGURE 5.15 – $GP^{FPCA}_{B\text{-splines}}$, modèle CCR : Les deux premières composantes principales. De haut en bas, pour une ACP appliquée à tous les coefficients de la base, et aux coefficients telle que $100p\%$ de l'énergie moyenne soit reconstituée (voir 3.4), avec $p = 1$ et $p = 0.99$. Le trait noir correspond au trait de côte.

Méthode	Temps (s) de la décomposition	Configuration ACPF	Proportion d'énergie (p), Nombre de coefficients ($\tilde{K}$)	Temps (s) de l'ACP
ACP				33
ACPF-ondellettes	1.7	1	$p = 1$, $\tilde{K} = 65\ 536$	33
		2	$p = 1$, $\tilde{K} = 35\ 830$	12.6
		3	$p = 0.99$, $\tilde{K} = 4\ 000$	1.5
ACPF-B-splines	299.6	1	$p = 1$, $\tilde{K} = 10\ 000$	3.2
		2	$p = 1$, $\tilde{K} = 5\ 451$	1.7
		3	$p = 0.99$, $\tilde{K} = 1\ 700$	0.7

Tableau 5.2 – Temps de calculs de l'ACP et de l'APCF pour le modèle du BRGM

Méthode	Temps (s) de la décomposition	Configuration ACPF	Proportion d'énergie (p), Nombre de coefficients ($\tilde{K}$)	Temps (s) de l'ACP
ACP				19.8
ACPF-ondellettes	2.1	1	$p = 1$, $\tilde{K} = 65\ 536$	19
		2	$p = 1$, $\tilde{K} = 10\ 771$	4
		3	$p = 0.99$, $\tilde{K} = 1\ 442$	1
ACPF-B-splines	295.5	1	$p = 1$, $\tilde{K} = 10\ 000$	3
		2	$p = 1$, $\tilde{K} = 1\ 725$	1.2
		3	$p = 0.99$, $\tilde{K} = 607$	0.5

Tableau 5.3 – Temps de calculs de l'ACP et de l'APCF pour le modèle de la CCR

Dans le cas des ondelettes, pour les deux modèles (voir les figures 5.12 et 5.13), on constate que les composantes principales des configurations 1 et 2 sont similaires à celles obtenues avec l'ACP standard. Dans le cas 2, l'ACP a été appliquée sur $\tilde{K} = 5\ 451$ coefficients d'ondelettes, pour le BRGM, et $\tilde{K} = 1\ 725$, pour la CCR. Cela représente respectivement une troncature de 91.7% et de 97.4% de la dimension de la base (voir les tableaux 5.2 et 5.3). Pour la configuration 3, dans le cas du BRGM, les deux composantes principales diffèrent au nord et nord-est des cartes, autour de la zone centrale de la carte (zone jaune/orange). Par exemple, là où la zone est jaune dans les cas de l'ACP, de 1 et de 2, la composante de 3 est verte (niveau 0, assimilé à la moyenne des HE). Dans le cas de la CCR, la structure de la deuxième composante obtenue avec la configuration 3, diffère légèrement des celles des autres configurations. Pour les deux premières configurations, on constate une légère augmentation de HE (environ entre 0.01 et 0.02) au-dessus de la zone bleu foncé, là où le niveau est à 0 (moyenne des HE) pour la configuration 3.

Dans le cas des B-splines, pour les deux modèles, on constate pour les configurations 1 et 2 que la structure spatiale des cartes sont similaires aux composantes principales de l'ACP standard. Mais contrairement aux ondelettes, l'amplitude des valeurs est différente. Dans le cas où 99% de l'énergie moyenne est conservée, l'amplitude des valeurs des composantes principales diffère de celles des autres configurations.

Dans les tableaux 5.2 et 5.3, on constate l'intérêt de réduire le nombre de variables de l'ACP, par troncature du nombre de coefficients. En effet, en fonction du nombre de coefficients, le coût calculatoire de l'ACP diminue. Cependant, pour l'ACPF, il faut aussi compter le temps de la décomposition des cartes sur la base de fonction. Pour les ondelettes, ce temps est négligeable (entre 1 et 2 secondes). Pour les B-splines, il est respectivement de 299.6 et 295.5 secondes pour le BRGM et la CCR (soit environ 5 minutes pour les deux modèles), en ayant parallélisé sur 5 cœurs, alors qu'un seul cœur est utilisé pour les ondelettes.

La diminution du nombre de coefficients, des coûts calculatoires et les différences acceptables entre les composantes principales (Figures 5.12-5.15) nous amènent à préconiser une ACPF ondelettes avec une énergie de 99%. Pour les splines, l'intérêt est aussi identifié avec une énergie de 99%. Cependant, le temps de calcul coûteux de l'orthonormalisation de la base est à souligner.

5.3.2 Précision de la prédiction

Les paramétrisations de $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$ et $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{B\text{-}splines}}$ sont faites comme dans la section 3.3.2 : par validation croisée à 10 blocs. Sur cette base, $n_{PC} = 2$ a été fixé pour les trois méthodes (voir la figure 5.9, pour $\mathrm{GP}^{\mathrm{PCA}}$). Les deux premières composantes principales correspondent à 96% de la variance expliquée pour les trois méthodes. Le nombre de coefficients utilisés dans l'ACP est choisi tel que 99% de l'énergie moyenne soit

conservée. Pour $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, cela correspond respectivement à $\tilde{K} = 4\ 000$ et $\tilde{K} = 1\ 442$ coefficients pour les modèles d'aléa du BRGM et de la CCR, soit une réduction respective de 93.9% et 97.8% en terme de nombre de variables. Pour $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, cela correspond respectivement à $\tilde{K} = 1700$ et $\tilde{K} = 607$ coefficients, soit une réduction respective de 97.4% et 99%. L'ACP dans $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$ et $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$ est donc appliquée à un nombre de variables radicalement inférieur à GP^{PCA}, ce dernier considérant le vecteur entier de taille $256 \times 256 = 65\ 536$.

On remarque que le nombre de coefficients sélectionnés pour le modèle d'aléa de la CCR, dans $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$et $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$, est respectivement presque 4 fois et 3 fois inférieur au nombre pour le modèle du BRGM. L'énergie des cartes estimées par la CCR (du moins 99% en moyenne) semble inférieure à celles du BRGM. En effet, dans la section 5.2, on avait déduit que les cartes de la CCR estimaient globalement des inondations moins étendues, en comparant la moyenne spatiale des HE. La figure 5.2 en donne quelques exemples.

Les HE sont des valeurs positives. Cependant, la prédiction par krigeage étant faite sur les composantes principales, des HE négatives peuvent être prédites. Si cela est effectivement le cas, les HE prédites comme négatives sont fixées à zéro.

Pour les trois méthodes, les Q^2 leave-one-out pour les modèles de krigeage des deux premières composantes principales sont donnés dans le tableau 5.4. Pour la CCR, les Q^2 semblent égaux pour les trois méthodes et sont très satisfaisant (supérieur à 0.9 pour les deux composantes). Pour le BRGM, les valeurs sont similaires avec des Q^2 très satisfaisant pour la première composante principale (0.93 pour les deux méthodes avec ACPF et 0.92 pour celle avec ACP). Les Q^2 de la seconde composante principale sont quand même satisfaisant avec une valeur de 0.79 pour la méthode avec ACP et 0.77 pour les deux avec ACPF.

	BRGM		CCR	
	PC1	PC2	PC1	PC2
GP^{PCA}	0.92	0.79	0.99	0.91
$\text{GP}^{\text{FPCA}}_{\text{wavelet}}$	0.93	0.77	0.99	0.91
$\text{GP}^{\text{FPCA}}_{\text{B-splines}}$	0.93	0.77	0.99	0.91

Tableau 5.4 – Pour les modèles BRGM et CCR : Q^2 leave-one-out pour les modèles de krigeage des deux premières composantes principales, notées PC1 et PC2, obtenues avec GP^{PCA}, $\text{GP}^{\text{FPCA}}_{\text{wavelet}}$, $\text{GP}^{\text{FPCA}}_{\text{B-splines}}$.

Dans un premier temps, la performance des trois méthodes est comparée en analysant la distribution des erreurs spatiales, mesurées par $RMSE(\mathbf{z})$ (voir (3.17)). Le RMSE est ici préféré au critère Q^2. En effet, il peut être exprimé en mètre, ce qui est plus facilement interprétable dans une perspective d'évaluation des risques. De plus, le Q^2 ne

peut être calculé pour les pixels qui sont constamment non inondés, le dénominateur étant égal à zéro. Les boxplots et les densités de probabilité estimées sont respectivement montrés dans les figures 5.17 et 5.16, en échelle logarithmique pour les boxplots.

Pour le BRGM, en regardant ces erreurs, on peut voir que les deux méthodes basées sur l'ACPF ($GP^{FPCA}_{wavelet}$et $GP^{FPCA}_{B\text{-}splines}$) surpassent celle basée sur l'ACP (GP^{PCA}), à la fois en moyenne et pour des valeurs extrêmes. Le premier quartile, la médiane, et le troisième quartile sont clairement plus petits pour les méthodes avec ACPF. De plus, les valeurs extrêmes (visibles sur les boxplots) sont limitées à 0.2m pour les méthodes avec ACPF, contrairement, à l'ACP qui elles peuvent atteindre 0.5m. Finalement, la $GP^{FPCA}_{wavelet}$est légèrement plus précise ici.

Pour la CCR, les différences ne sont pas si évidentes. D'ailleurs, les quantiles des boxplots de la figure 5.16, associés aux méthodes utilisant l'ACPF sont globalement supérieurs à ceux de la méthode avec ACP. Cependant, la différence entre les boxplots est fine. En effet, les quantiles des méthodes $GP^{FPCA}_{wavelet}$et $GP^{FPCA}_{B\text{-}splines}$sont supérieurs entre 5mm et 1cm à ceux obtenus en utilisant l'ACP. La précision des trois méthodes est mieux illustrée par la fonction de densité estimée (voir la figure 5.17). Pour les RMSE entre 0.03m et 0.15m, la densité des deux méthodes basées sur l'ACPF est supérieure à celle de l'ACP. Les valeurs les plus grandes des deux densités, dans cet intervalle, sont entre 0.03m et 0.075m, qui sont de faibles erreurs comparées au RMSE maximum atteint, qui est 0.5m. Néanmoins, les RMSE obtenus avec GP^{PCA}inférieurs à 0.03, ont une densité plus forte. Au vu de la fonction de densité associée à $GP^{FPCA}_{B\text{-}splines}$, la méthode semble être globalement moins précise que GP^{PCA}, contrairement au BRGM. Finalement, les résultats sont sensiblement similaires sur ce cas.

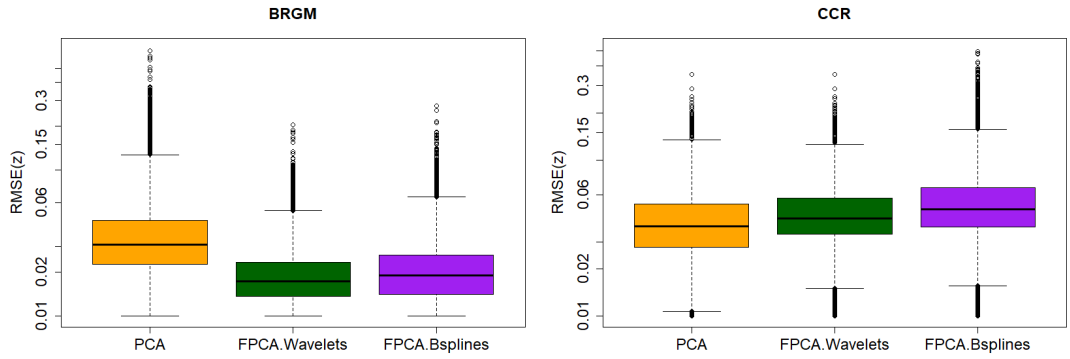

FIGURE 5.16 – Les boxplots des cartes RMSE des prédictions par GP^{PCA}, $GP^{FPCA}_{wavelet}$, et $GP^{FPCA}_{B\text{-splines}}$. À gauche, ceux obtenus à partir du modèle d'aléa du BRGM, à droite, ceux à partir de celui de la CCR. Le RMSE est tracé en échelle log base 10. Les RMSE égaux à zéro sont donc retirés de l'analyse. Étant des erreurs négligeables, les RMSE inférieurs à 1cm ne sont pas considérés dans les graphiques.

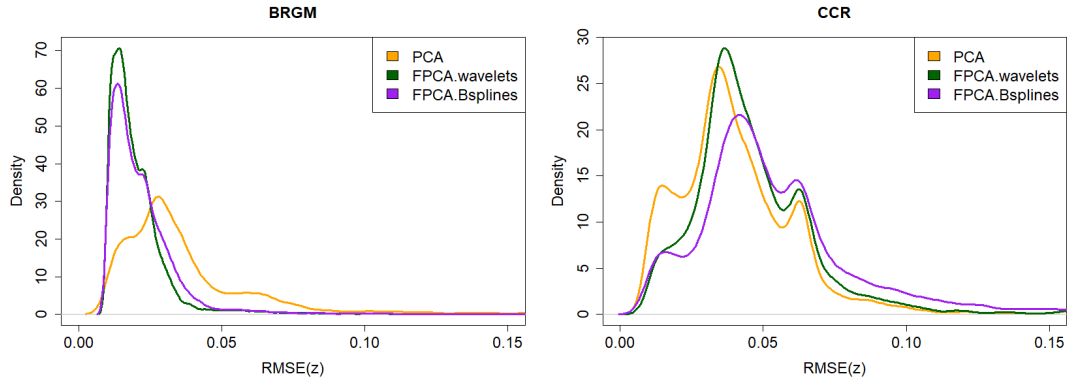

FIGURE 5.17 – Les estimations des fonctions de densité des cartes RMSE des prédictions par GP^{PCA}, $GP^{FPCA}_{wavelet}$, et $GP^{FPCA}_{B\text{-splines}}$. À gauche, ceux obtenus à partir du modèle d'aléa du BRGM, à droite, ceux à partir de celui de la CCR. Étant des erreurs négligeables, les RMSE inférieurs à 1cm ne sont pas considérés dans les graphiques.

Les densités et les boxplots dans les figures 5.17 et 5.16 donnent une information spatiale globale de la précision des prédictions. Cette dernière est analysée localement dans les figures 5.18 et 5.19, respectivement pour les modèles d'aléa du BRGM et de la CCR. Les cartes de RMSE obtenues avec les trois méthodes, y sont comparées.

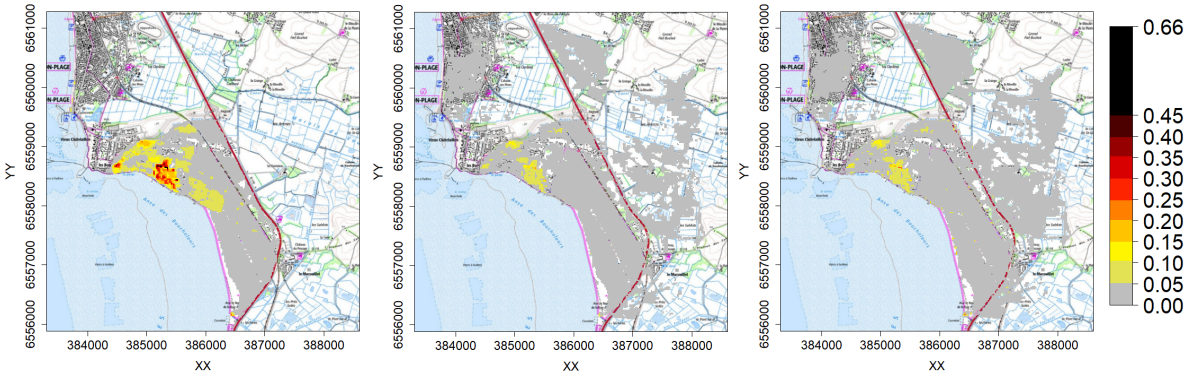

FIGURE 5.18 – BRGM : Les cartes de RMSE (en utilisant les 100 simulations de l'échantillon test) obtenues par GP^{PCA}(à gauche), $GP^{FPCA}_{wavelet}$(au milieu), et $GP^{FPCA}_{B\text{-}splines}$(à droite). Dans les cartes, les localisations sans aucune valeur de données correspondent aux localisations où le RMSE est strictement inférieur à 1cm, qui est une erreur négligeable.

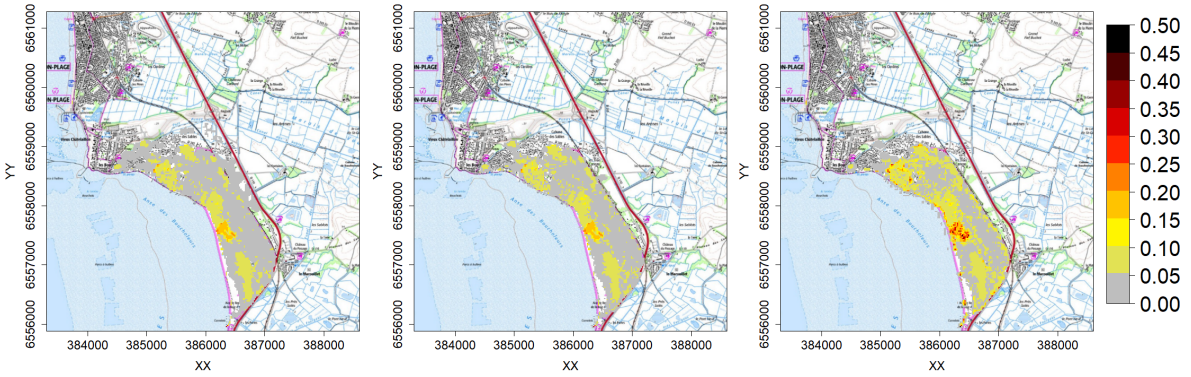

FIGURE 5.19 – CCR : Les cartes de RMSE (en utilisant les 100 simulations de l'échantillon test) obtenues par GP^{PCA}(à gauche), $GP^{FPCA}_{wavelet}$(au milieu), et $GP^{FPCA}_{B\text{-}splines}$(à droite). Dans les cartes, les localisations sans aucune valeur de données correspondent aux localisations où le RMSE est strictement inférieur à 1cm, qui est une erreur négligeable.

Pour le BRGM (voir la figure 5.18), pour GP^{PCA}, les erreurs les plus fortes sont localisées là où les plus fortes irrégularités sont observées dans la figure 5.2, c'est-à-dire dans la zone urbaine. Dans ces zones, les RMSE de $GP^{FPCA}_{wavelet}$et $GP^{FPCA}_{B\text{-}splines}$sont inférieurs à ceux de GP^{PCA}, par 0.1m et 0.2m. Cependant, en dehors de la zone centrale, les valeurs de RMSE des deux méthodes basées sur l'ACPF apparaissent légèrement supérieures à ceux de la carte associée à GP^{PCA}, mais pas plus de 0.05m, qui est une amplitude raisonnable. En effet, on constate que cette zone des deux composantes principales de

$GP^{FPCA}_{wavelet}$ et $GP^{FPCA}_{B\text{-splines}}$ ne détecte aucune variation par rapport à la moyenne des HE, lorsque seulement $p = 0.99$ de la proportion de l'énergie moyenne est conservée, contrairement à celles obtenues avec l'ACP standard.

Pour la CCR, pour les trois méthodes, on constate que les plus grandes erreurs sont situées au niveau du marais (zone orange). $GP^{FPCA}_{wavelet}$ génère une carte de RMSE légèrement moins étendue que celle obtenue avec GP^{PCA}. En effet, au nord de la zone centrale et du côté gauche de l'autoroute (ligne rouge du fond de carte), la carte associée à la méthode basée sur l'ACP a des erreurs entre 1cm et 5cm (les erreurs inférieures à 1cm n'étant pas tracées), alors que celle de $GP^{FPCA}_{wavelet}$ n'a aucune valeur, ce qui correspond à des erreurs entre 0cm et 1cm. Pour le reste de la carte, les erreurs semblent équivalentes. La structure de la carte obtenues avec $GP^{FPCA}_{B\text{-splines}}$ est similaire à celle de $GP^{FPCA}_{wavelet}$. Cependant à des pixels spécifiques (pixels rouges) dans la zone urbaine et du marais, les erreurs sont bien supérieures, avec des valeurs entre 0.25m et 0.5m.

Finalement, pour le BRGM, les méthodes utilisant l'ACPF semblent ici plus précises que GP^{PCA}. La décomposition sur base d'ondelettes donne des résultats plus précis que les B-splines. Pour la CCR, $GP^{FPCA}_{B\text{-splines}}$ est moins précise que GP^{PCA}(après interprétation de la figure 5.19). D'après les densités (voir la figure 5.17) et les cartes RMSE (voir la figure 5.19), GP^{PCA}et $GP^{FPCA}_{wavelet}$ont des résultats sensiblement similaires. Dans la prochaine section, les simulations pour faire l'analyse de sensibilité sont générées avec $GP^{FPCA}_{wavelet}$.

5.4 Analyse de sensibilité

Comme dans la section 4.3, une analyse de sensibilité (AS) a été faite pour les modèles d'aléa du BRGM et de la CCR, en remplaçant les simulateurs par un méta-modèle (obtenu avec $GP^{FPCA}_{wavelet}$, avec $\tilde{K} = 4\,000$, pour le BRGM, et $\tilde{K} = 607$, pour la CCR) entrainé sur les $n = 500$ simulations. Les indices de Sobol (total et du premier ordre) de chaque composante principale sont obtenus avec la procédure d'estimation de [Saltelli, 2002] (implémentée via la fonction sobolSalt du package sensitivity de **R**). Ils sont calculés à partir d'échantillons aléatoires Monte Carlo, de taille initiale $n_0 = 10^4$, et une loi uniforme est supposée pour chaque entrée (sur tout leur intervalle de variation, voir le tableau 5.1).

5.4.1 Analyse spatiale avec les composantes principales

Les indices de Sobol sont estimés pour les deux premières composantes principales générées à partir des simulations des modèles du BRGM et de la CCR. Ils sont montrés dans les figures 5.20 et 5.21.

La figure 5.20 donne les indices de Sobol estimés pour le modèle du BRGM. Le niveau 0 assimilé à la moyenne des HE correspond à la couleur bleu foncé. La première composante correspond à une augmentation entre 1cm et 2cm par rapport à la moyenne

sur l'ensemble de la zone rose. La variable la plus influente sur cette composante est l'amplitude de la marée T, qui a principalement de l'influence en interaction avec les autres paramètres. En effet, la différence entre l'indice total et l'indice du premier ordre est de 0.3. Néanmoins, T a aussi beaucoup d'influence seule (indice du premier ordre égal à 0.35). Les deux autres variables importantes sont S et t_0, qui ont elles aussi principalement de l'influence en interaction avec les autres paramètres. Elles n'ont que 10% d'influence seules (indice du premier ordre égale à 0.1). Les deux variables restantes (t_- et t_+) n'ont qu'une légère l'influence, en interaction avec les autres entrées.

La deuxième composante diffère de la première dans les zones principalement bleu clair et verte, qui correspond à une diminution entre 1cm et 4cm. La zone verte (avec quelques pixels jaunes et rouges) correspond à la zone urbaine. Sur cette composante, T est encore la variable la plus influente, mais d'intensité plus forte avec un indice total supérieur à celui de la première composante. À l'inverse, T a moins d'influence seule (indice du premier ordre inférieur à celui de la première composante). S et t_0 sont aussi les deux autres variables les plus influentes. Cependant, elles n'ont quasiment aucune influence seule (indice du premier ordre proche de 0), et ont donc de l'influence seulement en interaction avec les autres entrées. t_- et t_+ ont aussi peu d'influence que pour la première composante principale.

La figure 5.21 donne les indices de Sobol estimés pour le modèle de la CCR. Le niveau 0 assimilé à la moyenne des HE correspond à la couleur bleu clair. Les deux composantes principales montrent une augmentation de 2cm par rapport à la moyenne (zone bleu foncé). La principale différence entre les deux composantes est la zone correspondant au marais (rose, dans la première composante principale, et rouge, pour la deuxième). Dans la première composante principale, la zone montre une augmentation entre 4cm et 6cm par rapport à la moyenne. Dans la deuxième composante principale, elle correspond à une diminution de 6cm.

Sur les deux composantes, T est la variable la plus influente, avec un indice total presque égal à un. Sur la première composante, T a quasiment de l'influence à elle seule (indice total et indice du premier ordre sont presque égaux et proches de 1). t_0 et t_- ont une très légère influence (indice total inférieur à 0.05). Sur la deuxième composante, l'indice total est presque égal à 1. Mais l'indice du premier ordre étant inférieur, T a donc de l'influence avec les autres paramètres dans la deuxième composante. Deux autres paramètres influents sont t_0 et t_- avec des indices totaux à 0.2 et 0.18. Avec des indices du premier ordre à 0, elles ont de l'influence en interaction avec les autres entrées.

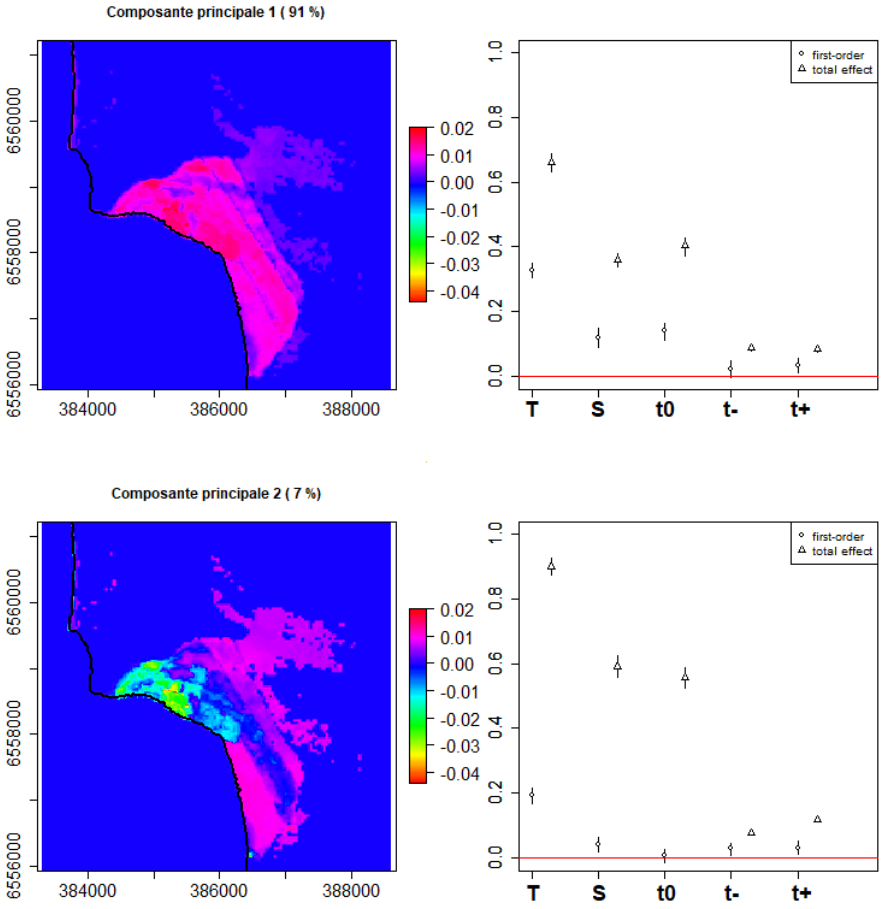

Figure 5.20 – À gauche, les deux premières composantes principales. À droite, les indices de Sobol estimés pour le modèle du BRGM. Les cercles correspondent aux indices du premier ordre, et les triangles, aux indices totaux.

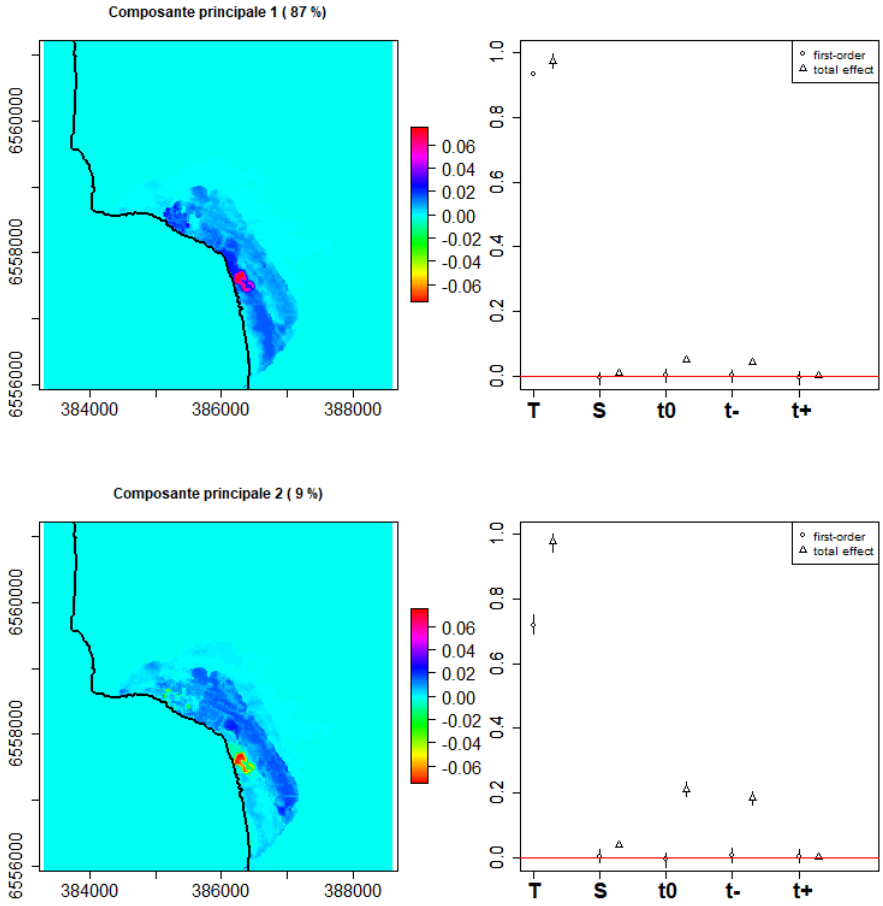

FIGURE 5.21 – À gauche, les deux premières composantes principales. À droite, les indices de Sobol estimés pour le modèle de la CCR. Les cercles correspondent aux indices du premier ordre, et les triangles, aux indices totaux.

5.4.2 Analyse de sensibilité avec l'indice généralisé

L'indice généralisé de sensibilité est montré dans la figure 5.22. L'amplitude de la marée T semble être le paramètre le plus influent pour les deux modèles, comme indiqué par l'indice du premier ordre le plus haut. Plus particulièrement, pour le modèle de la CCR, pour lequel les indices du premier ordre des autres paramètres sont estimés à zéro. Néanmoins, on constate une très légère influence de t_0 et t_- avec des indices totaux d'environ 0.05. Pour le modèle du BRGM, la différence entre l'indice total et celui du premier ordre indique que T a une forte interaction avec les autres variables d'entrée. Ses deux autres paramètres influents (de même importance) sont la surcote S et la différence de phase t_0. Ils ont essentiellement de l'influence avec les autres variables (les indices du premier ordre sont approximaticement 0.1, au lieu de 0.4 pour l'indice

total). Les deux paramètres restants (t_- et t_+) ont un effet négligeable, avec un indice total de 0.1 pour le deux.

Les résultats obtenus sont cohérents avec les nuages de points observés dans la figure 5.4. De plus, pour les deux modèles, ces résultats semblent physiquement cohérents avec le phénomène de débordement. Ne mettant en avant que T, l'analyse généralisé du modèle de la CCR a moins d'intérêt que l'analyse spatiale de la figure 5.21, où on peut voir l'influence non négligeable de t_0 et t_- sur la deuxième composante, correspondant tout de même à 9% de la variance expliquée.

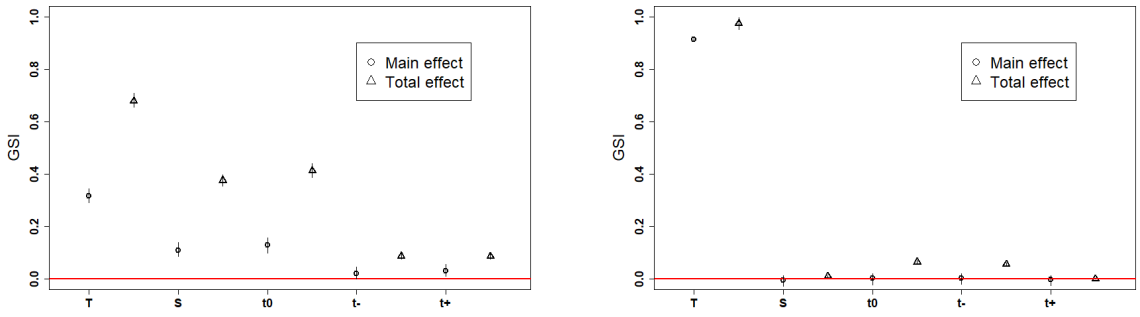

Figure 5.22 – Indices généralisés de sensibilité pour chaque entrée. À gauche, ceux du modèle d'aléa du BRGM. À droite, ceux du modèle d'aléa de la CCR. Les ronds représentent les indices du premier ordre. Les triangles représentent les indices totaux.

Finalement, ces résultats montrent que dans les deux cas, la submersion est fortement influencée par la marée. Les résultats diffèrent ensuite selon les modèles. Pour le BRGM, deux autres variables ressortent : l'amplitude de la surcote et le phasage de celle-ci avec le pic de marée, ce qui est logique. Pour le modèle CCR, cette importance de l'interaction entre la surcote et le phasage ne ressort pas de l'analyse. Une des explications est peut-être la très forte corrélation entre les hauteurs d'eau et l'amplitude de la marée, visible sur la figure 5.4.

5.5 Analyse de sensibilité : la tempête Xynthia

Dans cette section, l'intérêt de l'analyse de sensibilité est de se placer dans les conditions de fonctionnement d'un modèle d'aléa pour un événement réel (ici, la tempête Xynthia) et d'effectuer l'analyse de sensibilité de l'ensemble des paramètres qui peuvent influencer l'inondation liée à cet évènement.

5.5.1 Paramètres du modèle d'aléa

On s'intéresse ici à l'influence de l'ouverture de connexions hydrauliques sur l'estimation de l'inondation durant la tempête Xynthia (2010), sur le même site d'étude que précédemment. Des exemples de connexions hydrauliques sont les buses, les déversoirs, les brèches etc. Dans l'AS, on en considère sept, qui sont indiquées dans la figure 5.23 par des traits rouges. En entrée du modèle, sept variables binaires associées à chaque connexion sont considérées : avec 0 lorsque la connexion est fermée, et 1 lorsqu'elle est ouverte. Les variables associées aux connexions hydrauliques sont notées $conn_i$, avec $i = 1, \ldots, 7$.

D'autres paramètres du modèle sont aussi analysés : le forçage marin (niveau d'eau au large) et les coefficients de rugosité. Le BRGM et la CCR utilisent différents forçages marins. On considère en entrée du modèle une autre variable catégorielle à deux niveaux, prenant pour « valeur » "BRGM" ou "CCR". Cette variable indique lequel des deux forçages est utilisé dans le modèle d'aléa.

La rugosité caractérise la plus ou moins grande résistance d'un type de sol (béton, pelouse, sable,...) au passage de l'eau. La figure 5.23 est une carte des différents types d'occupation du sol considérés dans la zone des Bouchôleurs. La rugosité est généralement paramétrée en termes de coefficients de Manning ou de Strickler (inverse du coefficient de Manning). Ici, on considère les coefficients de Manning. Pour chaque type de sol, ce coefficient varie dans un intervalle. Le tableau 5.5 indique cet intervalle pour chaque coefficient. Chacun est noté comme indiqué dans ce tableau : frott.A, ..., frott.L. Pour l'analyse de sensibilité, on considère donc 8 variables catégorielles à deux niveaux, associées au forçage marin et aux 7 connexions hydrauliques, et 12 variables continues, correspondant aux coefficients de rugosité. La sortie du modèle est une carte estimant les hauteurs d'eau maximales atteintes durant la tempête Xynthia.

On présente les résultats pour les modèles d'aléa du BRGM et de la CCR.

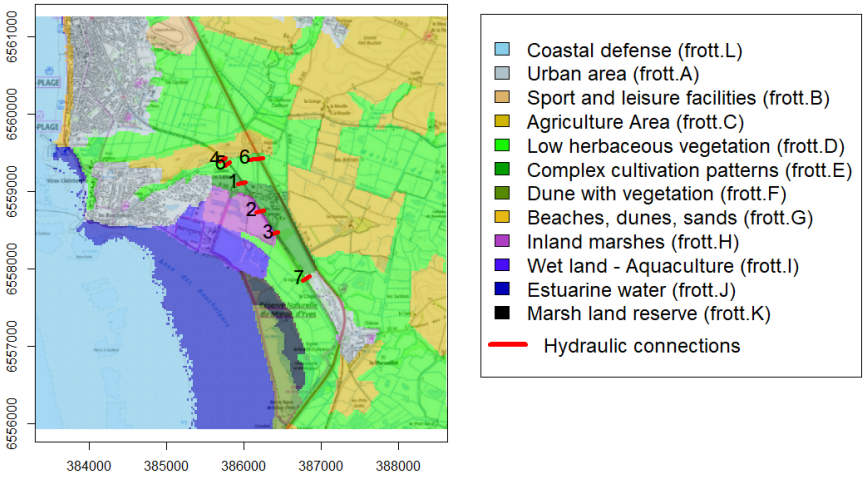

Figure 5.23 – Cartes des coefficients de rugosité. Les traits rouges correspondent aux connexions hydrauliques, numérotées de 1 à 7.

Notation	Type de sol	Intervalle (Coefficient de Manning)
frott.A	Zone urbaine	[0.169, 0.416]
frott.B	Installations sportives et de loisirs	[0.048, 0.074]
frott.C	Zone agricole	[0.044, 0.056]
frott.D	Végétation herbacée basse	[0.042, 0.048]
frott.E	Modèles de culture complexes	[0.042, 0.056]
frott.F	Dune avec végétation	[0.032, 0.038]
frott.G	Plages, dunes, sables	[0.032, 0.038]
frott.H	Marais intérieurs	[0.055, 0.105]
frott.I	Terres humides - Aquaculture	[0.038, 0.048]
frott.J	Eau estuarienne	[0.024, 0.037]
frott.K	Réserve marécageuse	[0.039, 0.051]
frott.L	Défense côtière	[0.015, 0.019]

Tableau 5.5 – Tableau des valeurs du coefficient de Manning (Entrées continues du modèle)

5.5.2 Analyse de sensibilité

Afin d'obtenir les simulations nécessaires à l'AS, un méta-modèle $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$ a été construit. La méthode de construction est ici la même que précédemment mais contrairement à la section 5.3, les modèles de krigeage sur chaque composante principale ont en entrée des variables catégorielles (voir la section 2.1.5).

5.5.2.1 Modèle du BRGM

Le méta-modèle $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$ a été construit à partir d'un échantillon d'apprentissage constitué de $n = 768$ simulations du modèle d'aléa du BRGM. Le plan d'expériences de ces simulations a été construit tel que chacune des $2^8 = 256$ combinaisons des niveaux des variables catégorielles apparaissent 3 fois. Il a été complété par un plan hypercube latin de taille n pour définir les valeurs des 12 entrées continues. $\tilde{K} = 10\ 896$ coefficients d'ondelettes ont été utilisés dans l'ACP pour réduire la dimension des cartes (le nombre de pixels est $K = 65\ 536$). $\tilde{K}$ correspond à environ 100% de l'énergie moyenne. Les trois premières composantes principales représentent au total 96% de la variance expliquée. Les Q^2 du leave-one-out des modèles de krigeage sur l'échantillon d'apprentissage de taille n sont 0.88, 0.92 et 0.78, pour les trois premières composantes principales. Ces trois composantes sont représentées dans la figure 5.24. Une analyse de sensibilité spatiale a été faite à partir de celles-ci, en analysant leurs structures spatiales et en estimant les indices de Sobol associés (voir la figure 5.25). Les indices sont calculés à partir d'échantillons aléatoires Monte Carlo de taille initial $n_0 = 10^3$, et une loi uniforme est supposée pour chaque entrée.

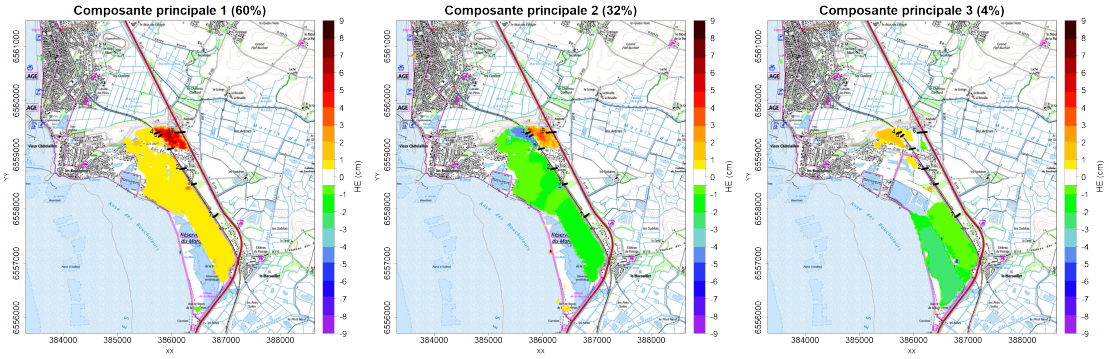

FIGURE 5.24 – De gauche à droite, les trois premières composantes principales obtenues à partir des cartes estimées par le modèle d'aléa du BRGM. Les traits noirs correspondent aux connexions hydrauliques.

Le niveau 0 des composantes principales (voir la figure 5.24) est associé à la moyenne des HE, et correspond aux zones où il n'y a aucune couleur. Les indices de Sobol donnés dans la figure 5.25 montrent que les seules variables avec de l'influence sur les deux premières composantes principales sont le forçage marin et les connexions hydrauliques

4 et 5. Contrairement aux variables Forçage et $conn_5$, l'influence de la connexion 4 est faible, avec des indices égaux à 0.05 sur la première composante, et égaux à 0.03 sur la seconde. Sinon, le paramètre le plus influent est le forçage marin sur les deux premières composantes principales. Cependant, l'effet de $conn_5$ n'est pas négligeable avec des indices du premier ordre et totaux respectivement à 0.38 et 0.43, sur la première composante, et environ 0.45 pour les deux, sur la seconde.

Pour expliquer l'influence de $conn_5$, on peut analyser les structures spatiales de la Figure 5.24. On remarque que :

— la **première composante principale** : la zone jaune à l'entrée de $conn_4$ et $conn_5$, qui correspond à une augmentation entre 5mm et 1cm du niveau d'eau par rapport à la moyenne, et la zone rouge à la sortie de $conn_4$ et $conn_5$, qui correspond à une augmentation plus forte du niveau d'eau, entre 5cm et 9cm.

— la **deuxième composante principale** : la zone bleue, à l'entrée de $conn_4$ et $conn_5$ et la zone jaune/orange, à la sortie de $conn_4$ et $conn_5$. Ces zones peuvent représenter un écoulement à travers ces deux connexions hydrauliques. En effet, la zone bleue correspond à une diminution entre 4cm et 5cm du niveau d'eau, et la zone rouge, à une augmentation entre 1cm et 4cm.

D'après les observations sur les deux premières composantes principales, qui représentent 92% de la variance expliquée, la propagation de l'eau sur les terres semble fortement influencée par la connexion 5, en plus du forçage marin.

Néanmoins, quelques coefficients de rugosité ont aussi une légère influence sur la troisième composante, où les connexions hydrauliques n'ont aucune influence. Le paramètre le plus influent est le coefficient associé aux eaux estuariennes (frott.J), dont l'indice du premier ordre est égal à 0.37. Le second est le coefficient associé à la réserve marécageuse (frott.K), dont l'indice du premier ordre est égal à 0.12. Les autres paramètres influents sont frott.F (dune avec végétation) et frott.G (plages, dunes, sables). Mais, ils ont principalement de l'influence avec les autres coefficients de rugosité (indice du premier ordre à zéro, mais les indices totaux sont respectivement 0.13 et 0.10). frott.J et frott.K ont plus d'influence en interaction avec d'autres variables avec des indices totaux à 0.65 et 0.47. On remarque deux zones vertes sur la troisième composante principale (une plus claire que l'autre), qui correspondent à une diminution par rapport à la moyenne entre 1cm et 3.5cm. Ces zones correspondent à celles des coefficients frott.K et frott.F. Cependant il faut remarquer que la 3ème composante principale représente à peine 4% de la variance expliquée. Par conséquent, l'influence des coefficients de rugosité est bien plus faible que le forçage et $conn_5$.

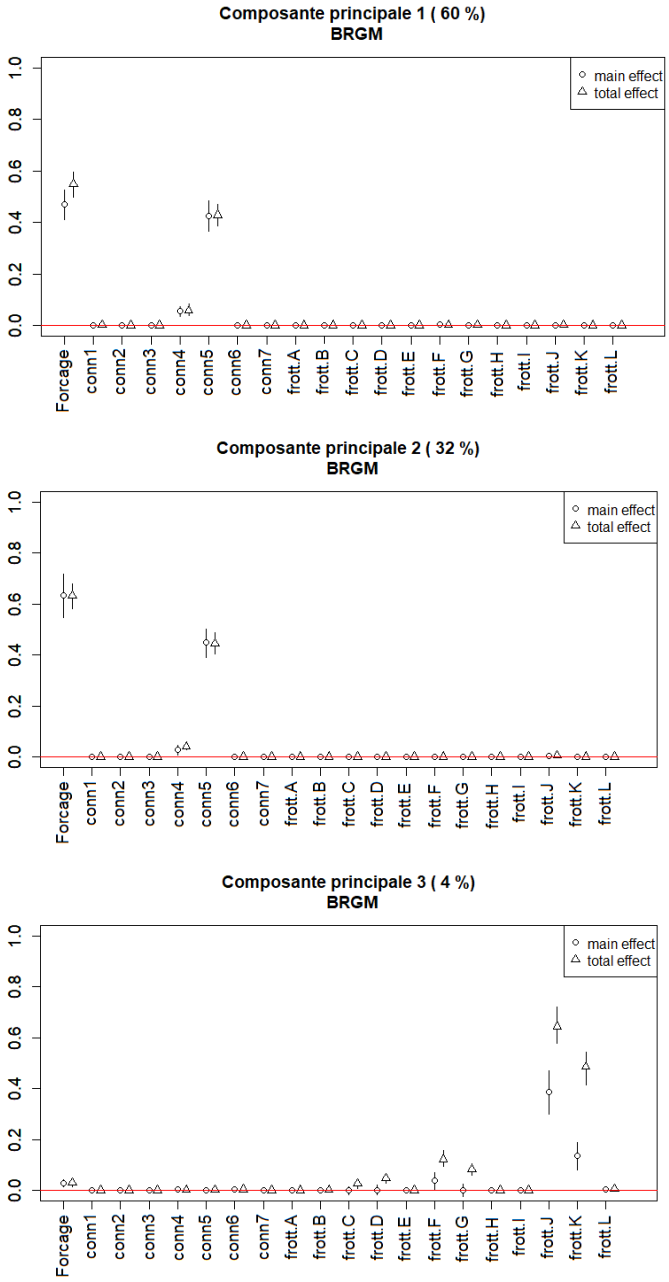

FIGURE 5.25 – De haut en bas, les indices de Sobol estimés sur les trois premières composantes principales. Les cercles correspondent aux indices du premier ordre. Les triangles correspondent aux indices totaux. Les traits noirs représentent l'intervalle de confiance à 95% de des estimations. Les variables associées aux connexions hydrauliques sont notées $conn_i$, avec $i = 1, \ldots, 7$.

Les indices généralisés de sensibilité ont été estimés à partir des indices montrés dans la figure 5.25 et sont montrés dans la figure 5.26. Le forçage et $conn_5$ sont les entrées avec

les plus grands indices, qui sont respectivement 0.50 et 0.40. On observe aussi la faible influence de $conn_4$.

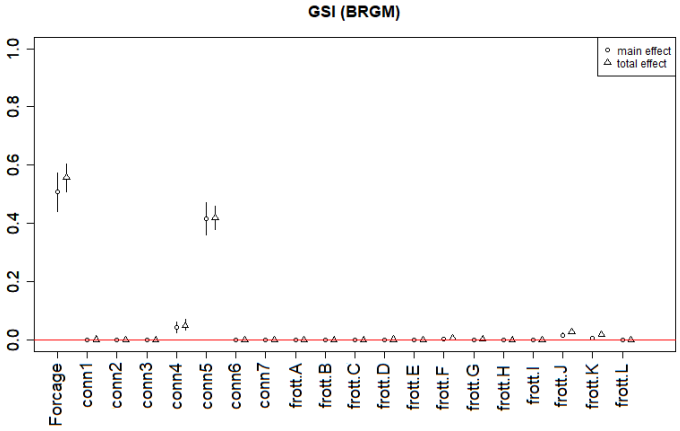

Figure 5.26 – Modèle BRGM : Indices généralisés de sensibilité des paramètres de forçage, de connexions hydrauliques, et des coefficients de rugosité.

Finalement, pour le modèle BRGM, les paramètres influençant le plus l'estimation de l'inondation lors de la tempête Xynthia (2010) est le forçage marin et la cinquième connexion hydraulique.

5.5.2.2 Modèle de la CCR

Le plan d'expérience de l'échantillon d'apprentissage du modèle de la CCR a été construit selon la même procédure que celui du BRGM. $\tilde{K} = 11\ 421$ coefficients d'ondelettes ont été utilisés dans l'ACP pour réduire les dimensions des cartes (nombre de pixels des cartes : $K = 65\ 536$). De même que le BRGM, $\tilde{K}$ correspond à environ 100% de l'énergie moyenne. L'analyse de sensibilité est faite à partir des deux premières composantes principales. Les Q^2 du leave-one-out des modèles de krigeage sur l'échantillon d'apprentissage de taille $n = 768$ sont 0.99 et 0.97, pour les deux premières composantes principales.

Les figures 5.27 et 5.28 montrent les deux premières composantes principales, et les indices de Sobol (total et du premier ordre) associés. Sur la première composante principale, le paramètre le plus influent est le forçage marin avec un indice de Sobol du premier ordre à 0.86. Les indices de Sobol des autres paramètres sont égaux (ou quasiment égaux) à zéro. Dans la figure 5.27, la carte associée (celle de gauche) met en avant la zone jaune/orange, qui correspond à une augmentation entre 0cm et 3cm des HE par rapport à la moyenne. Ce qui serait donc principalement expliqué par le forçage. Le pourcentage de variance expliquée de la première composante principale est 88%. L'influence du forçage l'emporte donc sur celles des variables les plus influentes sur les autres composantes principales, dont les variances expliquées sont inférieures ou

égales à 4%.

Sur la seconde composante principale (4% de variance expliquée), la variable la plus influente est frott.A avec un indice du premier ordre estimé à environ 0.97. frott.A est le coefficient de rugosité de la zone urbaine. La zone rouge/orange/jaune de cette composante principale (voir la figure de droite de 5.27) correspond à cette zone. Les couleurs du jaune au rouge sont associées à une augmentation moyenne entre 0cm et 6cm des HE par rapport à la moyenne. On constate que les couleurs forment un dégradé allant du rouge au jaune, ce qui pourrait illustrer l'écoulement des eaux du littoral à l'intérieur des terres.

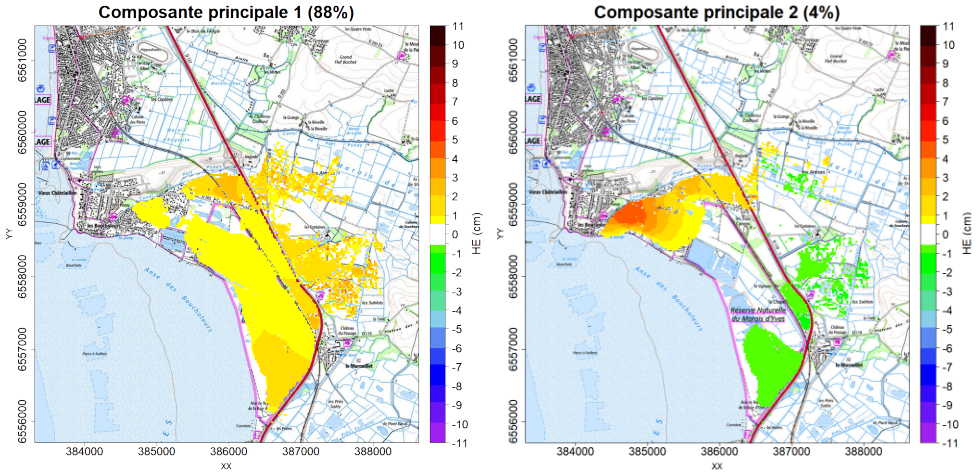

FIGURE 5.27 – De gauche à droite, les deux premières composantes principales obtenues à partir des cartes estimées par le modèle d'aléa de la CCR.

Les indices généralisés de sensibilité ont été estimés à partir des indices de Sobol estimés sur les composantes principales, et sont montrés dans la figure 5.29. On observe que le forçage est le paramètre le plus influent, avec un indice de sensibilité du premier ordre égale à 0.83. frott.A, qui est très influent sur la 2^{de} composante principale, a des indices de sensibilité du premier ordre et total faibles (environ égaux à 0.05). En effet, le pourcentage de variance expliquée de cette composante principale est 4%. Contrairement au BRGM, les connexions hydrauliques n'ont aucune influence.

Finalement, pour le modèle CCR, le forçage est le paramètre le plus influent sur l'estimation de l'inondation lors de la tempête Xynthia. Néanmoins, il est à noter que même si son indice généralisé est faible, frott.A a une forte influence sur la 2^{de} composante principale.

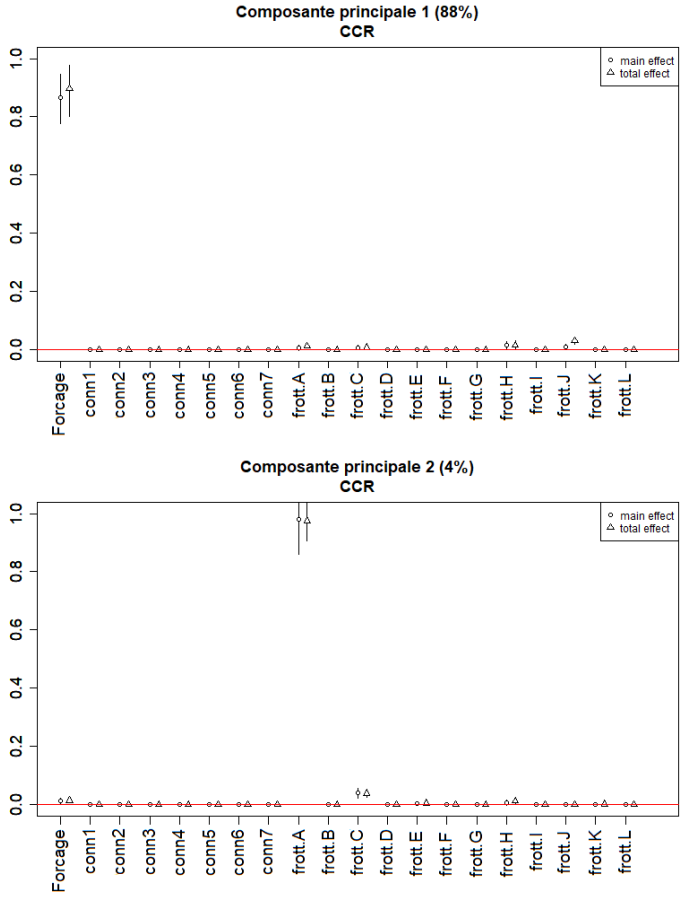

Figure 5.28 – De haut en bas, les indices de Sobol estimés sur les deux premières composantes principales. Les cercles correspondent aux indices du premier ordre. Les triangles correspondent aux indices totaux. Les traits noirs représentent l'intervalle de confiance à 95% des estimations.

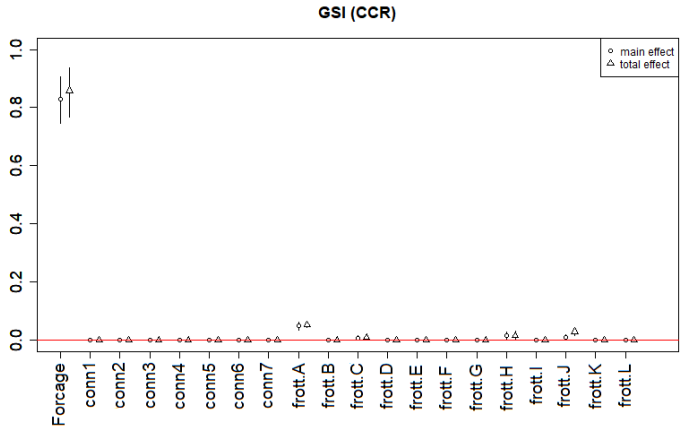

Figure 5.29 – Modèle CCR : Indices généralisés de sensibilité des paramètres de forçage, de connexions hydrauliques, et des coefficients de rugosité.

5.5.2.3 Comparaison des deux modèles

Pour les deux modèles, les estimations des inondations lors de la tempête Xynthia sont fortement influencées par le forçage marin. Le modèle BRGM est aussi influencé par la 5^e connexion hydraulique (voir la figure 5.23). D'après l'indice généralisé, les coefficients de rugosité ont une influence négligeable comparée à celles du forçage et de $conn_5$. Mais frott.K et frott.J ont néanmoins une influence non négligeable sur la troisième composante principale, correspondant quand même à 4% de la variance expliquée. Le modèle CCR semble plus sensible au coefficient de rugosité de la zone urbaine (voir les indices de Sobol de la 2^{de} composante principale de la figure 5.28).

Finalement, pour les deux modèles, les résultats montrent l'intérêt d'avoir un forçage marin le plus précis possible. Pour le BRGM, on peut voir l'importance de considérer les connexions hydrauliques.

5.6 Perspective sur la combinaison des deux modèles d'aléa

Les deux analyses de sensibilité précédentes ont montré que les deux modèles d'aléa ont des comportements différents. En effet, l'étendue et la structure spatiale des estimations sont différentes. De plus, les paramètres d'entrées influents ne sont pas les mêmes selon le modèle utilisé.

Une perspective est de développer une méthode associant les simulations des modèles d'aléa du BRGM et de la CCR. Le temps de calcul du modèle BRGM est coûteux (entre 30 minutes et 1 heure pour une seule estimation) et plus long que celui de la CCR (environ 5 minutes). L'objectif est d'améliorer les estimations d'un modèle à partir des informations de l'autre. Une idée est de considérer les sorties des deux simulateurs comme celles d'un même modèle, et d'explorer les méthodes de méta-modélisation pour les modèles multi-sorties. On pourrait mettre en œuvre une technique de co-krigeage ([Chiles and Delfiner, 1999, Wackernagel, 2013]). Une alternative est de concaténer les simulations faites par les deux modèles d'aléa, et d'ajouter une variable catégorielle en entrée du modèle, qui indique quel modèle est utilisé en prenant pour valeur « BRGM » ou « CCR ». On considère donc le modèle suivant :

$$\begin{array}{rccl} f & : & \{BRGM, CCR\} \times \mathcal{X} & \rightarrow & \mathbb{L}^2(\mathcal{Z}) \\ & & \boldsymbol{\omega} = (\mathbf{u}, \mathbf{x}) & \mapsto & y_{\boldsymbol{\omega}}(\mathbf{z}) \end{array} \tag{5.7}$$

avec $\mathbf{u}$ une variable catégorielle binaire spécifiant quel modèle d'aléa est utilisé (c'est-à-dire $BRGM$ ou CCR), et $\mathcal{X}$ l'espace des entrées des deux modèles. Dans la section 5.5, $\mathcal{X}$ correspond à $\{\mathrm{F}_{BRGM}, \mathrm{F}_{CCR}\} \times \{0,1\}^7 \times \prod_{i=1,\dots,12}[a_i, b_i]$, où $\mathrm{F}_{\mathbf{mod}}$ est le forçage marin estimé, avec $\mathbf{mod} \in \{BRGM, CCR\}$, et $[a_i, b_i]$, avec $i = 1, \dots, 12$, l'intervalle des valeurs des 12 coefficients de rugosité frott.A, ..., frott.L. Durant la

thèse, la deuxième approche a été testée. Cependant, la méthodologie n'a pas été approfondie et pourrait faire l'objet de futurs travaux.

Les deux modèles ont estimé l'inondation des Bouchôleurs, lors de la tempête Xynthia, en variant le forçage marin utilisé, l'ouverture des connexions hydrauliques et les valeurs des coefficients de rugosité (voir la section 5.5.1). Le temps de calcul du modèle BRGM étant plus long que celui de la CCR, moins de simulations sont disponibles pour le BRGM que la CCR. Deux configurations des plans d'expériences (notés PE) des modèles d'aléa du BRGM et de la CCR sont testées :

— PE emboîtés : Tout le plan d'expérience du modèle BRGM est inclus dans celui de la CCR.
— PE différents : Les plans d'expériences des deux modèles diffèrent (aucune des simulations des deux modèles n'ont les mêmes entrées continues, c'est-à-dire les coefficients de rugosité).

Quelle que soit la configuration du plan d'expériences, l'échantillon d'apprentissage du modèle de la CCR contient 1 280 observations, et celui du BRGM est de taille 256 :

— Pour la CCR : Chacune des $2^8 = 256$ combinaisons des niveaux des variables catégorielles apparaît 5 fois. Le PE est complété par un hypercube latin de taille $n_{ccr} = 256 \times 5 = 1\,280$.
— Pour le BRGM : Les combinaisons des niveaux des variables catégorielles n'apparaissent qu'une fois. Le PE est complété par un hypercube latin de taille $n_{brgm} = 256$.

À partir des observations de l'échantillon d'apprentissage, un méta-modèle $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$ est construit. Il est utilisé à la place du modèle (5.7) pour prédire les sorties d'un échantillon test de taille 512 pour le BRGM, et 1 536 pour la CCR. $\tilde{K} = 4036$ et $\tilde{K} = 4110$ coefficients d'ondelettes sont respectivement sélectionnés pour les cartes de sortie des PE emboîtés et des PE différents. Dans les deux cas, $\tilde{K}$ correspond à 99.5% de l'énergie moyenne. Pour les deux configurations de PE, $n_{PC} = 2$ composantes principales sont estimées. Elles correspondent à 99.3% de variance expliquée pour des PE emboîtés, et à 99.1% pour différents PE.

En pratique, il n'y a pas de cartes de hauteurs d'eau réelles que l'on peut comparer avec les simulations. Cependant, il est possible d'utiliser la base de données de sinistres de la CCR pour valider l'inondation simulée. Les sinistres sont des biens (habitations ou entreprise), qui ont été endommagés lors de Xynthia et pour lesquels l'assuré a reçu une indemnisation de la part de son assureur. On peut penser que l'aléa est de bonne qualité s'il détecte un grand nombre de sinistre en évitant de détecter à tort des biens non inondés. Plusieurs indices existent : le POD (Probabilité de détection), le POFD (Probabilité de Fausse Détection), le TSS (True Skill Score) et le CSI (Critical Sucess Index) [Schaefer, 1990]. Ces scores sont calculés à partir des données de 4 966 biens assurés dans les Bouchôleurs, pour lesquels l'état de l'inondation (inondé ou sec) est

connu durant Xynthia.

Le tableau 5.6 définit l'état des biens dans les ensembles A, B, C, et D. On définit les hypothèses H_0 et H_1 comme suit :
— H_0 : le bien est inondé.
— H_1 : le bien n'est pas inondé.
« Vrai » désigne la réalité et « accepté » désigne le résultat du modèle.

	Hypothèse H_0, vraie	**Hypothèse H_1, vraie**
Hypothèse H_0, acceptée	Vrai positif (A)	Faux positif (C)
Hypothèse H_1, acceptée	Faux négatif (B)	Vrai négatif (D)

Tableau 5.6 – Tableau de contingence croisant les hypothèses H_0 et H_1.

a, b, c et d correspondent respectivement aux cardinaux de A, B, C, et D. Les scores sont définis par les formules données par (5.8)

$$\begin{gathered} \mathrm{POD} = \frac{a}{a+b} \qquad \mathrm{POFD} = \frac{c}{c+d} \\ \mathrm{TSS} = \mathrm{POD} - \mathrm{POFD} \quad \mathrm{CSI} = \frac{a}{a+b+c} \end{gathered} \tag{5.8}$$

Tous les scores présentés sont entre 0 et 1. Le POD correspond à la probabilité qu'un sinistre soit bien détecté, c'est-à-dire qu'une inondation soit à la fois observée et estimée. Le POFD correspond à la probabilité qu'un sinistre soit faussement détecté, c'est-à-dire qu'une inondation soit estimée mais non observée. On cherche donc à ce que le POD soit au plus proche de 1, et le POFD au plus proche de 0. Le TSS est un score global de la qualité de la prédiction, et prend en compte le POD et le POFD. Le CSI correspond à la proportion de sinistres en commun entre l'observation et l'estimation. Plus le TSS ou le CSI sont proches de 1, plus l'estimation de l'aléa est considérée pertinente. Ces scores sont mesurés pour chaque carte estimée.

La figure 5.30 montre les boxplots des scores de l'ensemble des cartes estimées par les deux modèles d'aléa et $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$. L'amélioration ou la dégradation des scores est similaire pour les deux modèles. On observe une augmentation globale du POD en associant les observations des deux modèles. Cependant, le POFD augmente aussi. Néanmoins, les valeurs du POFD restent faibles et proches de 0. En effet, pour le BRGM et la CCR, les médianes sont respectivement 0.084 et 0.021, pour différents PE, et 0.084 et 0.020, pour des PE emboîtés. Pour les scores globaux TSS et CSI, on constate une amélioration du TSS, mais une dégradation du CSI. Néanmoins, pour la CCR, le CSI reste supérieur à 0.77. L'estimation reste donc pertinente. Cependant, pour le BRGM, le CSI est globalement entre 0.60 et 0.63 lorsque les observations des

deux modèles sont associées, celui du modèle BRGM est globalement supérieur à 0.67.

À propos de considérer des PE différents ou emboîtés, on constate que les scores sont légèrement moins bons pour le POD, le TSS et le CSI, lorsque des PE emboîtés sont considérés. L'amélioration (ou la dégradation) des scores a néanmoins un comportement similaire lorsque des PE différents sont considérés. Cependant, pour valider cette comparaison des deux configurations du PE, des tests statistiques doivent être réalisés (par exemple : validation croisée, bootstrap [Efron and Tibshirani, 1994], etc.).

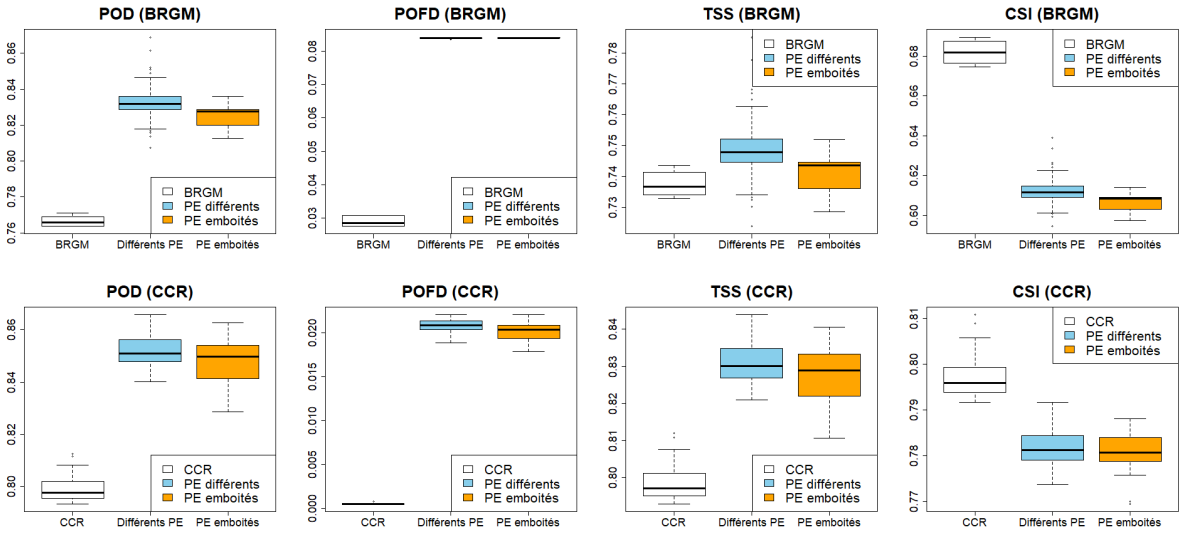

Figure 5.30 – De gauche à droite, boxplots des scores POD, POFD, TSS, et CSI des estimations des modèles et du méta-modèle. En bleu clair, les scores obtenus avec des plans d'expériences différents, en orange, ceux obtenues avec des plans d'expériences emboîtés (tout le plan d'expérience du modèle BRGM est inclus dans celui du modèle CCR). En haut, les scores de l'aléa du BRGM. En bas, les scores de l'aléa de la CCR.

Finalement, associer les échantillons d'apprentissage des deux modèles augmente le nombre de détection de sinistre (bâtiment inondé). Cependant, cela augmente aussi le nombre de fausses détections, c'est-à-dire qu'un bien soit estimé inondé alors qu'il ne l'est pas. En effet, les cartes estimées par $\mathrm{GP}^{\mathrm{FPCA}}_{\mathrm{wavelet}}$ estiment des inondations plus étendues que les modèles d'aléa (voir des exemples de cartes dans la figure 5.31). En termes de performance globale, l'estimation semble globalement s'améliorer, avec une augmentation du TSS. Cependant, la proportion de sinistre en commun se dégrade (voir CSI de la figure 5.30), mais l'estimation reste néanmoins pertinente pour la CCR.

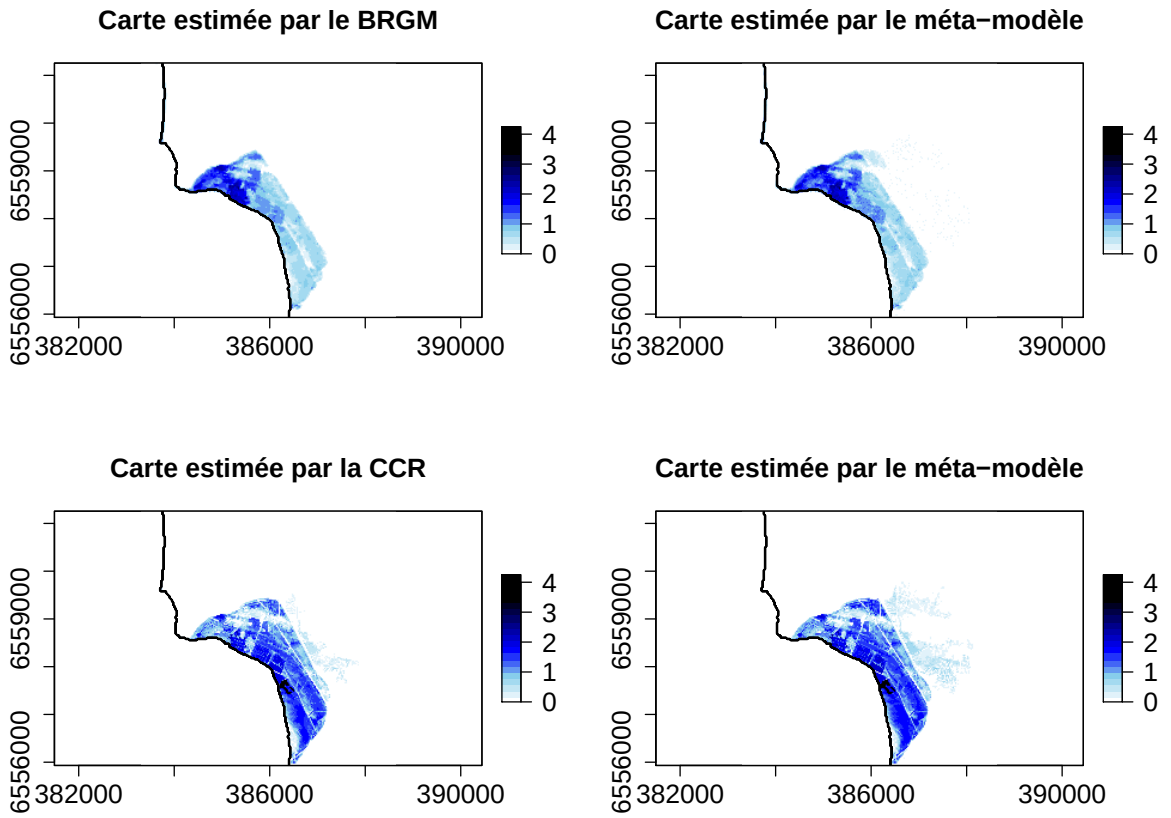

FIGURE 5.31 – À gauche, des exemples de cartes estimées par le BRGM et la CCR. À droite, les cartes estimées par le méta-modèle pour les mêmes entrées. Le trait noir correspond au trait de côte.

Quatrième partie

Contribution logicielle : Package R

Chapitre 6

Le package GpOutput2D

6.1 Introduction

L'ensemble des travaux a fait l'objet d'un développement d'un package **R** [R Core Team, 2020], appelé GpOutput2D. L'objectif du package est de construire des méta-modèles pour les modèles avec une sortie fonctionnelle bidimensionnelle, en utilisant les méthodes de régression par processus gaussien (voir la section 2.1). Les fonctions du package proviennent du travail de la thèse présenté dans le chapitre 3, et contiennent les étapes de l'algorithme 1. Les fonctions principales sont implémentées par des méthodes S3 et sont présentées dans le tableau 6.1.

Des outils d'analyse de l'ACP fonctionelle et de la prédiction du méta-modèle ont aussi été implémentés. La fonction plot permet de tracer :

- les fonctions propres,
- la proportion d'énergie moyenne de chaque coefficient de la base de fonctions utilisée (B-splines ou ondelettes),
- des barplots représentant le pourcentage de variance expliquée de chaque composante principale,
- le nombre de coefficients sélectionnés en fonction de la proportion d'énergie moyenne fixée (p).

La fonction error.predict mesure la précision de la prédiction du méta-modèle en calculant la matrice des RMSE et des Q^2.

Le package GpOutput2D a été développé et partagé sur GitHub (voir https://github.com/tranvivielodie/GpOutput2D). La vignette du package est donnée dans la section 6.2. Elle a été écrite en anglais.

Nom de la fonction	Description
Fpca2d	Appliquer une ACPF à des données fonctionnelles bidimensionnelles.
km_Fpca2d	Construire des modèles de krigeage sur chaque score obtenu avec la fonction **Fpca2d**, en utilisant la fonction **km** du package **DiceKriging**.
gp_Fpca2d	Construire des modèles de krigeage sur chaque score obtenu avec la fonction **Fpca2d**, en utilisant la fonction **gp** du package **kergp**.
predict	Prédiction de la sortie fonctionnelle bidimensionnelle de la fonction objective en un nouveau point, en utilisant les modèles de krigeage sur les scores de l'ACPF.

Tableau 6.1 – Fonctions principales du package GpOutput2D.

6.2 Vignette du package GpOutput2D

GpOutput2D : An R Package for metamodelling models with Two-Dimensional functional Output by using FPCA and Gaussian Process Regression Models

PERRIN Tran Vi-vi Élodie

Contents

1 Introduction

The aim of GpOutput2D package is to build metamodels for models with two-dimensional functional output, by using Gaussian Process (GP) regression methods [Williams and Rasmussen, 2006] (also called kriging models). The following simulator is considered:

$$\begin{array}{rcccc} f & : & \Omega \subseteq \mathbb{R}^d & \rightarrow & \mathbb{L}^2(\mathcal{Z}) \\ & & \mathbf{x} & \mapsto & y_{\mathbf{x}}(\mathbf{z}) \end{array} \tag{1}$$

We assume that we know $n \in \mathbb{N}^\star$ simulations of f: $\{(\mathbf{x}_i, y_i(\mathbf{z})), i = 1, \ldots, n\}$, with $y_i(.) = y_{\mathbf{x}_i}(.)$. We aim at predicting the map $f(\mathbf{x}^\star)$ for a new point $\mathbf{x}^\star$.

The functions of GpOutput2D come from the PhD work of [Perrin et al., 2021]. The main lines of the method are:

1. To reduce the infinite dimension of $\mathbb{L}^2(\mathcal{Z})$ to a finite dimensional space of size $K \in \mathbb{N}^\star$, by representing the functional output by an orthonormal basis functions, which is denoted $\Phi(\mathbf{z}) = (\phi_1(\mathbf{z}), \ldots, \phi_K(\mathbf{z}))^\top$:

$$y_{\mathbf{x}}(\mathbf{z}) = \sum_{k=1}^{K} \alpha_k(\mathbf{x})\phi_k(\mathbf{z}) = \boldsymbol{\alpha}(\mathbf{x})^\top \Phi(\mathbf{z}) \tag{2}$$

with $\boldsymbol{\alpha}(\mathbf{x})$ the coefficients vector of $y_{\mathbf{x}}(.)$ on $\Phi(\mathbf{z})$.

2. To truncate the number of basis coefficients, by keeping those which influence the most the energy.
3. To apply a standard multivariate PCA to the selected coefficients.
4. To build GP models for each principal component score (or coordinate).
5. To predict the scores of $y_{\mathbf{x}^\star}(.)$ on principal components, by using their associated kriging models.
6. To predict $y_{\mathbf{x}^\star}(.)$ by embedding the estimated scores into the initial functional space: $\mathbb{L}^2(\mathcal{Z})$.

An other **R** package exists, called FPCA2D, with functions for performing FPCA on two-dimensional functional data (even for three-dimensional functional data, with FPCA3D). However, the only implemented basis functions is the Fourier basis. Furthermore, we do not have control over the Fourier decomposition. In GpOutput2D, other basis functions have been implemented: two-dimensional wavelet and B-splines basis. B-splines basis is not orthonormal. Then, it can be orthonormalized by performing Gram-Schmidt method [Björck, 1994], or specific procedures for splines [Liu et al., 2019, Qin, 2000, Redd, 2012].

Wavelet basis is commonly used in image processing [Mallat, 1999]. Indeed, a key advantage over Fourier transform is the capture of both frequency and location information. Spline functions are piecewise polynomials. They are commonly chosen for approximation of non-periodic functional data [Ramsay, 2006, Ramsay and Silverman, 2007]. Basis system have been developed for spline functions. The B-splines basis has been used in the package. GpOutput2D depends on waveslim package, for two-dimensional wavelet transform, and on orthogonalsplinebasis package, for two-dimensional B-splines basis (and for its orthonormalization).

To approximate at best the functional data, the basis dimension K may be high dimensional. Therefore, a preliminary selection step is added, by choosing the $\tilde{K}$ basis terms which are most influential based on the energy decomposition (see [Perrin et al., 2021]). Then, PCA is applied to them. The non-selected basis terms are estimated by empirical mean. With this idea, the method can be performed on large dimensional vectors, which contain basis coefficients.

The construction of GP models, for each principal component score, is based on km of DiceKriging package, or gp of kergp package. Both functions fit kriging model on data. The difference is that kergp lets the user to define his own covariance kernel. In particular, building a GP metamodel is possible for models which have categorical input variables [Roustant et al., 2020]. However, kergp is a laboratory package, and may evolve in its future. Therefore, GpOutput2D also depends on DiceKriging, for user which are interested in stable software.

The main functionalities of GpOutput2D are implemented as S3 methods. They are shown in Table 1.

Method Name	Description
Fpca2d	Applying FPCA to two-dimensional functional data.
km_Fpca2d	Building a kriging models on each score obtained by using **Fpca2d**, with the **km** function from **DiceKriging** package.
gp_Fpca2d	Building a kriging models on each score obtained using **Fpca2d**, with the **gp** function from **kergp** package.
predict	Prediction of the two-dimensional functional output of the objective function at a new point using GP model on scores of FPCA.

Table 1: Main functions of GpOutput2D package.

2 An analytical test case

Here, we illustrate how GpSpatialOutput works on a 2D toy example. The toy function, called Campbell2D, has eight scalar inputs ($d = 8$) and a spatial map as output (e.g. a function which depends on two inputs ($\mathbf{z} = (z_1, z_2)$) corresponding to spatial coordinates). The Campbell2D function was first introduced by Marrel et al. [2010]. (3) shows the model, and (4) is the formula of the spatial output.

$$\begin{array}{rccc} f & : & [-1,5]^8 & \to & \mathbb{L}^2([-90,90]^2) \\ & & \mathbf{x} = (x_1, \ldots, x_8) & \mapsto & y_{\mathbf{x}}(\mathbf{z}) \end{array} \tag{3}$$

where $\mathbf{z} = (z_1, z_2) \in [-90, 90]^2$, $x_j \in [-1, 5]$ for $j = 1, \ldots, 8$, and

$$\begin{array}{rl} y_{\mathbf{x}}(z_1, z_2) = & x_1 \exp\left[-\frac{(0.8z_1+0.2z_2-10x_2)^2}{60x_1^2}\right] + (x_2 + x_4) \exp\left[\frac{(0.5z_1+0.5z_2)x_1}{500}\right] + \\ & x_5(x_3 - 2) \exp\left[-\frac{(0.4z_1+0.6z_2-20x_6)^2}{40x_5^2}\right] + \\ & (x_6 + x_8) \exp\left[\frac{(0.3z_1+0.7z_2)x_7}{250}\right] \end{array} \tag{4}$$

This function has been implemented in GpSpatialOutput. For the application, the spatial domain $[-90, 90]^2$ is discretized on an uniform grid of dimension $n_{\mathbf{z}} \times n_{\mathbf{z}}$, with $n_{\mathbf{z}} = 64$.

```
> # inputs of the Campbell2D function
> x1<-rep(-1,8); x2<-rep(5,8); x3<-c(5,3,1,-1,5,3,1,-1)
> X <- rbind(x1,x2,x3) # inputs
> #
> # spatial domain
> nz<-64 # nz^2 is the size of the spatial domain
> z<-seq(-90,90,length=nz) # spatial coordinates
> #
> # Campbell2D function
> Y = Campbell2D(X,z,z) # outputs
```

We use the raster package to plot spatial data. A rotation matrix function, called rot90, will be performed on matrix, in order to get maps in the right rotation in a raster object.

```
> library(raster) # Geographic Data Analysis
> #
```

```
> # a raster model
> rst <- raster(xmn = -90, xmx = 90, ymn = -90, ymx= 90,
+               nrows=nz,ncols=nz)
> #
> # matrix rotation function
> rot90 <- function(x){(apply(t(x), 2, rev))}
```

Figure 1 shows examples of Campbell2D outputs. The output map presents strong spatial heterogeneities, sometimes with sharp boundaries. Furthermore, the spatial distribution is different according to the $\mathbf{x}$ values.

```
> # Colors associated to z values
> couleur <- colorRampPalette(c("blue","green",
+                               "yellow","orange","red"))
> #
> # Plot of three examples of Campbell2D output map.
> par(mfrow=c(1,3))
> #
> for(i in 1:3){
+   x<- get(paste("x",i,sep="")) # input name
+   rst[]<-rot90(Y[,i]) # raster model
+   #
+   # Plot of the Campbell2D output map
+   plot(rst,col=couleur(100),legend.width=2)
+   #
+   # Plot title : input vector associated to the output map
+   xstr <- paste(x,collapse = " , ")
+   xtitle <- paste(c("x=(",xstr,")"),collapse=" ")
+   title(paste("Campbell2D output for \n",xtitle)) # title
+ }# end for i
```

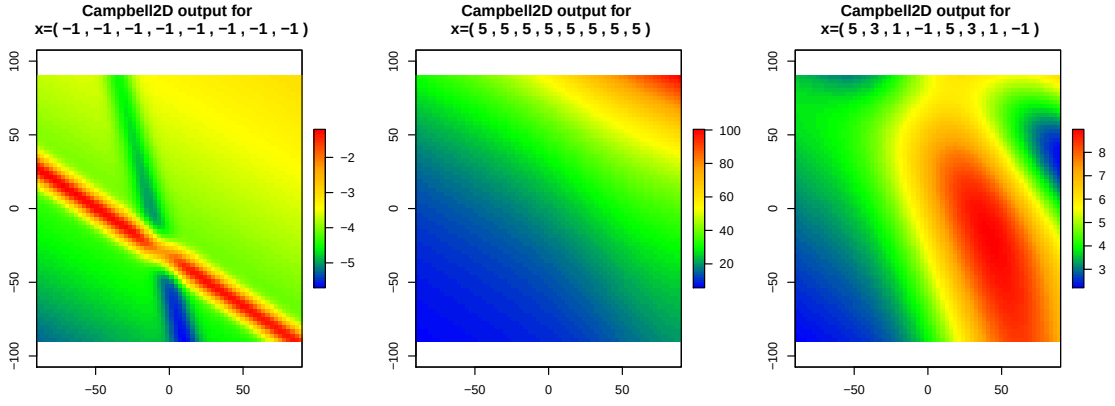

Figure 1: Example of Campbell2D outputs. From left to right, inputs are $\mathbf{x} = (-1, -1, -1, -1, -1, -1, -1, -1)$, $\mathbf{x} = (5, 5, 5, 5, 5, 5, 5, 5)$, and $\mathbf{x} = (5, 3, 1, -1, 5, 3, 1, -1)$.

3 Functional Principal Component Analysis (FPCA)

Fpca2d is a wrapper function to perform FPCA on two-dimensional functional data (images, maps, etc.), given a projection method. The two implemented functional basis are : orthonormal B-splines and wavelet.

To illustrate how to use FPCA, a learning sample of size $n = 200$ is considered, with a space-filling design of experiment (doe), which is constructed by using Latin Hypercube Sampling (LHS) design, and optimized by the SA algorithm [Dupuy et al., 2015], (implemented on the DiceDesign R package).

```
> library(lhs) # Latin Hypercube Sample
> library(DiceDesign) # Design of Computer Experiments
> #
> # design of experiment
> n<-200 # size of the learning sample
> doe <- maximinLHS(n=n,k=8) # LHS design
> doe_learn <-maximinSA_LHS(doe)$design # optimized design
> #
> # range of Campbell2D inputs
> rg <- c(-1,5)
> doe_learn<-doe_learn*(rg[2]-rg[1]) + rg[1]
> #
> # Output Spatial domain
> nz<-64
> z <- seq(-90,90,length=nz)
> #
> # Campbell2D output
> Y<-Campbell2D(doe_learn,z,z)
```

3.1 Functional basis decomposition

GpOutput2D depends on waveslim package for wavelet decomposition, by using dwt.2d function [Mallat, 1999, Vetterli and Kovacevic, 1995].

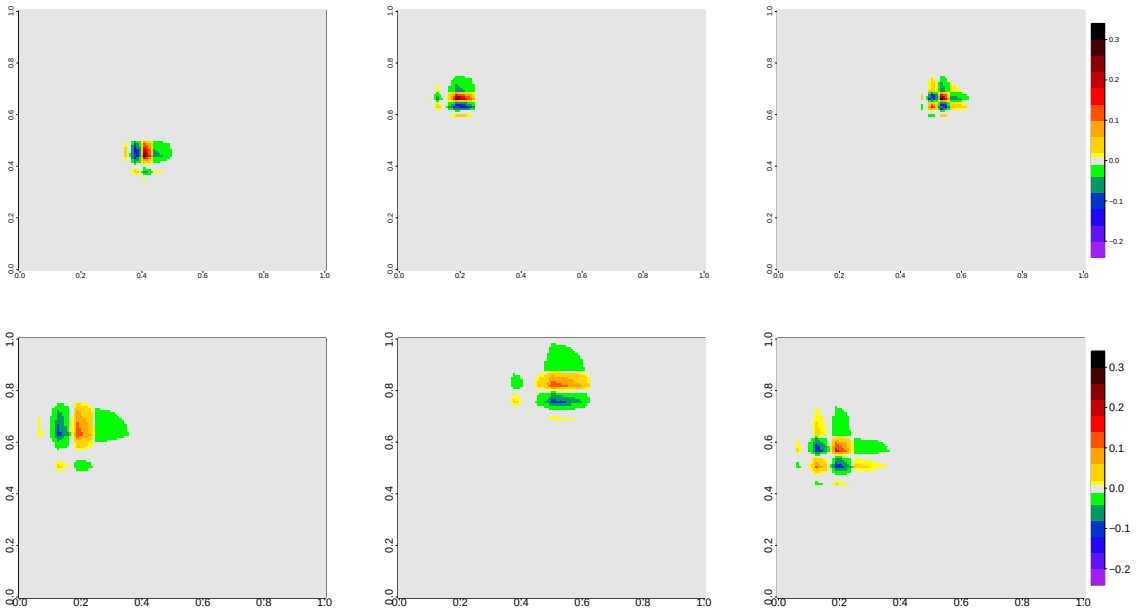

Figure 2: Examples of D4 Daubechies wavelets on $[0,1]^2$, which is discretized into a grid of size 128×128. From left to right, the figures represent examples of horizontal, vertical and diagonal wavelets. The top figures are wavelets at scale 3. The bottom figures are wavelets at scale 4.

In GpOutput2D, we also propose to use B-splines basis to approximate data. However, the truncation of the number of coefficients for PCA needs to work on orthonormal basis [Perrin et al., 2021]. Therefore, OrthoNormalBsplines2D builds a B-splines basis, which is orthonormalized according to a given orthogonalization method. The available methods are "GS", for the Gram-Schmidt method [Björck, 1994], and "Redd", for a specific method for B-splines, which is based on matrix representation of the functional basis [Redd, 2012]. We detail here how OrthoNormalBsplines2D builds the B-splines basis.

The orthonormalized B-splines is build as follows :

1. Two 1D B-splines basis are built for each dimension of the spatial domain. They are denoted $\boldsymbol{\psi}(\mathbf{z}) = (\psi_1(\mathbf{z}), \ldots, \psi_K(\mathbf{z}))$ and $\boldsymbol{\psi}'(\mathbf{z}) = (\psi'_1(\mathbf{z}), \ldots, \psi'_{K'}(\mathbf{z}))$, with $K, K' \in \mathbb{N}$.
2. $\boldsymbol{\psi}(\mathbf{z})$ and $\boldsymbol{\psi}'(\mathbf{z})$ are orthogonalized. If ortho = "GS", the standard Gram-Schmidt method is performed by using the gramSchmidt function from the pracma package. If ortho = "Redd", the ortogonalization is performed by using the orthogonalsplinebasis package. The method consists to represent the B-splines basis in matrix form, and to perform a matrix product with the root of the Gram matrix $\mathbf{G} = (\int \psi_k(\mathbf{z})\psi'_k(\mathbf{z})d\mu(\mathbf{z}))_{1 \leq k,k' \leq K}$ (see [Redd, 2012]). The new basis are respectively denoted $\boldsymbol{\psi}_\perp(\mathbf{z})$ and $\boldsymbol{\psi}'_\perp(\mathbf{z})$. The default orthogonalization method is "GS".
3. $\boldsymbol{\psi}_\perp(\mathbf{z})$ and $\boldsymbol{\psi}'_\perp(\mathbf{z})$ are then normalized :

$$\tilde{\boldsymbol{\psi}}(\mathbf{z}) = \frac{\boldsymbol{\psi}_\perp(\mathbf{z})}{||\boldsymbol{\psi}_\perp(\mathbf{z})||_2} \quad \text{and} \quad \tilde{\boldsymbol{\psi}}'(\mathbf{z}) = \frac{\boldsymbol{\psi}'_\perp(\mathbf{z})}{||\boldsymbol{\psi}'_\perp(\mathbf{z})||_2}$$

4. We consider the 2D basis which is obtained by tensorization,

$$\phi_{k,l}(\mathbf{z})) = \tilde{\boldsymbol{\psi}}_k(\mathbf{z})\tilde{\boldsymbol{\psi}}'_l(\mathbf{z}),\ k = 1, \dots K,\ l = 1, \dots, K'$$

The following code show how to build an orthonormal B-splines basis.

```
> #=======================================
> # To build an orthonormal B-splines basis
> #=======================================
>
> z.knots <-seq(-90,90,length=35)# knots for the B-splines basis
> OPhi_GS <- OrthoNormalBsplines2D(z,z,z.knots,z.knots,ortho="GS")
> OPhi_Redd <- OrthoNormalBsplines2D(z,z,z.knots,z.knots,ortho="Redd")

> #===============
> # Raster model
> #===============
>
> library(raster)
> # raster model
> rst <- raster(xmn=-90,xmx=90,ymn=-90,ymx=90,
+               nrows=64,ncols=64)
> # rotation matrix function
> rot90 <- function(x){apply(t(x),2,rev)}

> #=======================
> # plot functions of OPhi
> #=======================
>
> # indices of the OPhi basis
> k<-c(10,20,30)
> l<-c(5,15,25)
> par(mfrow=c(2,3),mar=(c(3,4,4,2)+0.1))
> # with ortho="GS"
> for(i in 1:3){
+   rst[]<- rot90(OPhi_GS[,k[i],,l[i]])
+   plot(rst, col = rainbow(128), horizontal=TRUE,
+        zlim=c(-0.19,0.681),legend.width=1.5,
+        main="orthogonalization \n with Gram-Schmidt")
+ }# end for i
> # with ortho="Redd"
> for(i in 1:3){
+   rst[]<- rot90(OPhi_Redd[,k[i],,l[i]])
+   plot(rst, col = rainbow(128), horizontal=TRUE,
+        zlim=c(-0.19,0.681),legend.width=1.5,
+        main="orthogonalization \n with orthogonalsplinebasis")
+ }# end for i
```

Figures 3 and 4 show examples of orthonormal basis functions, respectively by using ortho ='GS', and ortho ='Redd'. Basis functions seems similar. However, after performing a substraction of both basis (see

Figure 5), we see a difference in range.

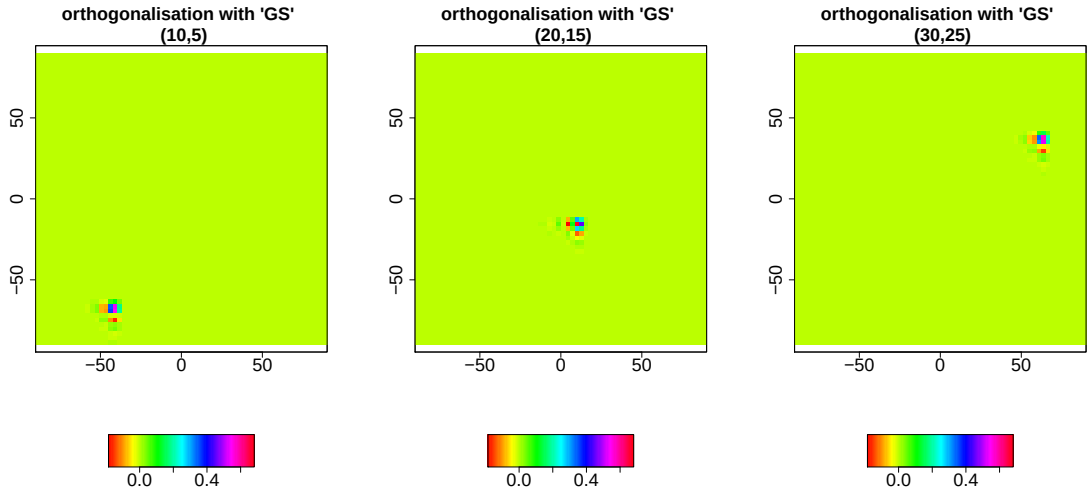

Figure 3: Example of functions of the orthonormal B-splines basis, which are obtained with the Gram-Schmidt method. From left to right, coordinates in the B-splines basis are : $(10, 5)$, $(20, 15)$, $(30, 25)$.

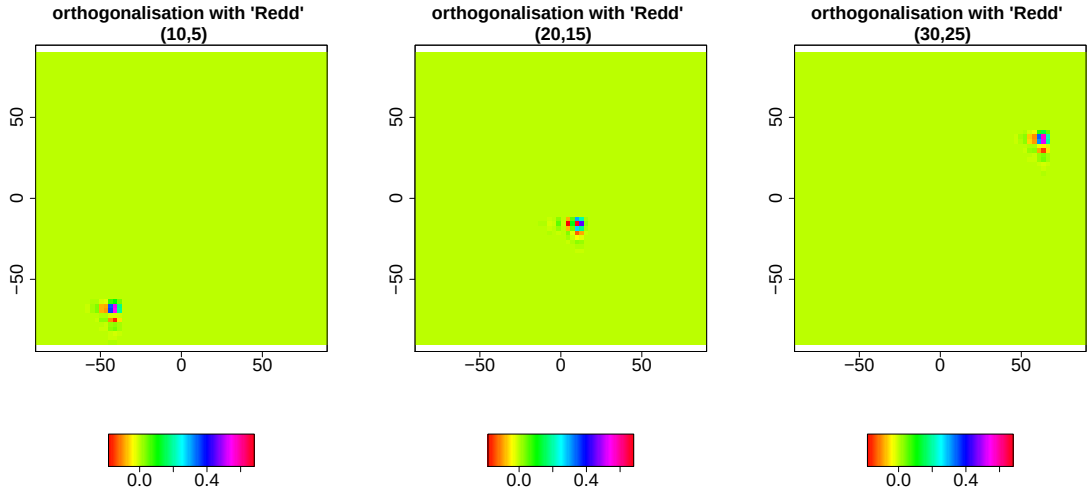

Figure 4: Example of functions of the orthonormal B-splines basis, which are obtained with the orthogonalsplinebasis package. From left to right, coordinates in the functional basis are : $(10, 5)$, $(20, 15)$, $(30, 25)$.

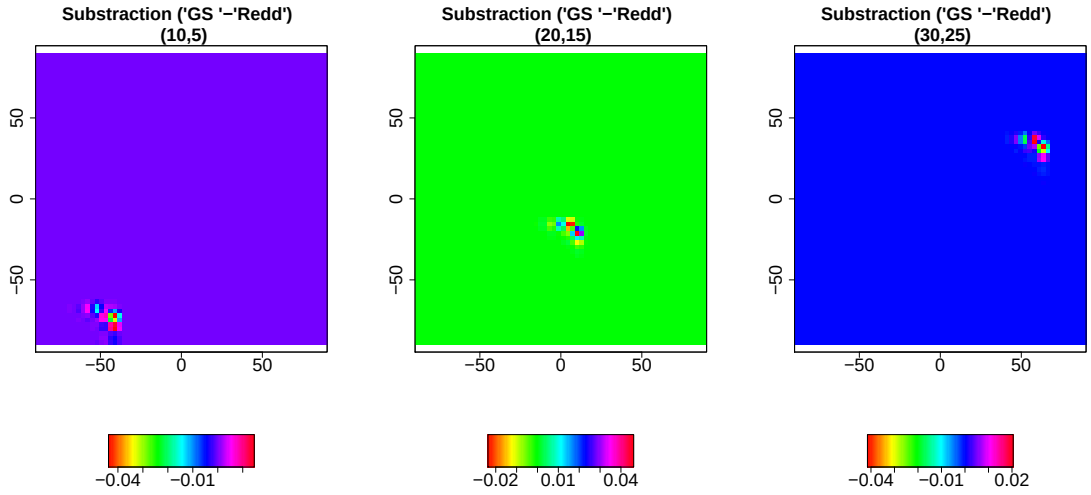

Figure 5: Substraction of orthonormal B-splines basis functions obtained by 'GS' and by 'Redd'. From left to right, coordinates in the functional basis are : $(10, 5)$, $(20, 15)$, $(30, 25)$.

3.2 Application of FPCA

According to Ramsay [2006], after performing wavelet or orthonormal B-splines decompositions, FPCA is equivalent to apply a standard multivariate PCA on the coefficients. In order to speed up FPCA, standard PCA is applied on the most "important" coefficients, which are selected according to their associated mean part of energy, as in Perrin et al. [2021]. Therefore, in Fpca2d, the total mean proportion of energy, called p, can be fixed. The number of coefficients is then calibrated according to its value. Otherwise, instead of giving p, the number of coefficients can be directly given. The basis parameter allows to choose the projection method : "Wavelet", for wavelet basis, and "Bsplines", for orthonormal B-splines. For orthogonalizing B-splines basis, the default method is Gram-Schmidt (ortho = "GS").

```
> # parameters for B-splines
> K<-35
> z.breaks <- seq(-90,90,length=K) #knots fot B-splines
> norder<-2 # B-splines order
> # parameter for wavelet decomposition
> J<-1# depth of the decomposition
> wf<-"d4" # type of wavelets, here Daubechies 4
> # number of principal components
> nPC <- 5
> #### FPCA ####
>
> # using wavelets
> FPCA_dwt <- Fpca2d(method = "Wavelets", x=Y,
+                    J=J,wf=wf, # wavelet parameters
+                    rank.=nPC,ncoeff=4000)
```

```
> # using Bsplines
> FPCA_bs <- Fpca2d(method = "Bsplines", x=Y,
+                     z1=z, z2=z, z1.knots=z.breaks, z2.knots=z.breaks,
+                     norder=2, # B-splines parameters
+                     rank.=nPC,p=1)
```

A S3 method has been implemented, in order to plot characteristics of FPCA. Different plots can be done according to the specified type : c("inertia", "energy", "MeanPoe", "eigenfunctions"). For "inertia" and "eigenfunctions", the argument PC allows to specify the plotted principal components by giving a vector with theirs numbers. The parameter "inertia" gives a barplot (see Figure 7), which illustrates the proportion of explained variance of each principal component. "eigenfunctions" gives the eigenfunction images (see Figure 6). "MeanPoe" corresponds to an image of coefficients (from wavelet or B-splines basis) mean proportion of energy (see Figure 8). The parameter "energy" gives a barplot (see Figure 9), which shows the number of coefficients used on PCA according to the total mean proportion of energy. The given percentages correspond to the percentage of selected coefficients.

```
> par(mfrow=c(1,3))
> plot(z1=z,z2=z,FPCA_dwt,type="eigenfunctions",PC=1:3)

> plot(FPCA_dwt,type="inertia",PC=1:nPC)

> plot(FPCA_dwt,type="MeanPoe")

> plot(FPCA_dwt,type="energy",p=seq(0.5,1,by=0.05))
```

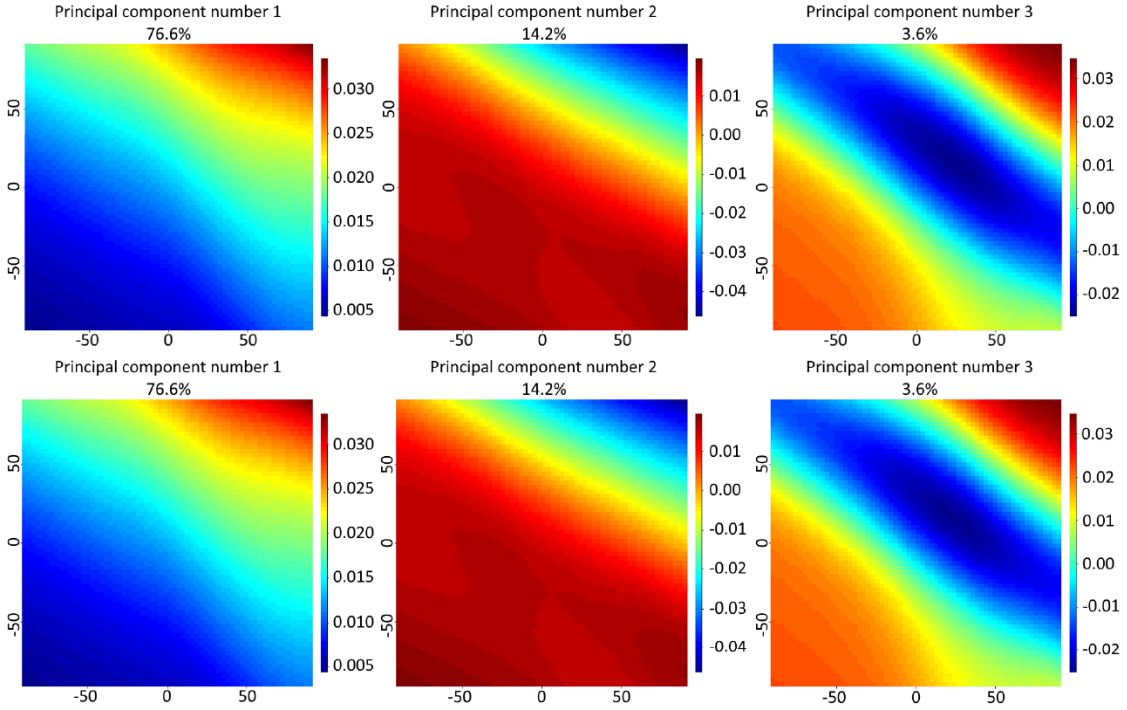

Figure 6: From left to right, the three first eigenfunctions, which correspond respectively to $76,6\%$, 14.2%, and 3.6% of the explained variance (same for both lines). First line correspond to the eigenfunctions of FPCA by using wavelet basis. The second line correspond to those obtained by using an orthonormal B-splines basis (with ortho ='GS').

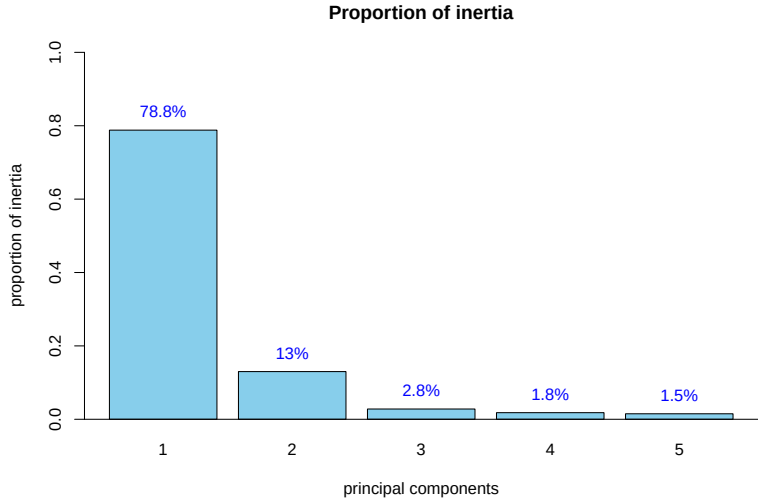

Figure 7: Proportion of explained variance of the five first principal components.

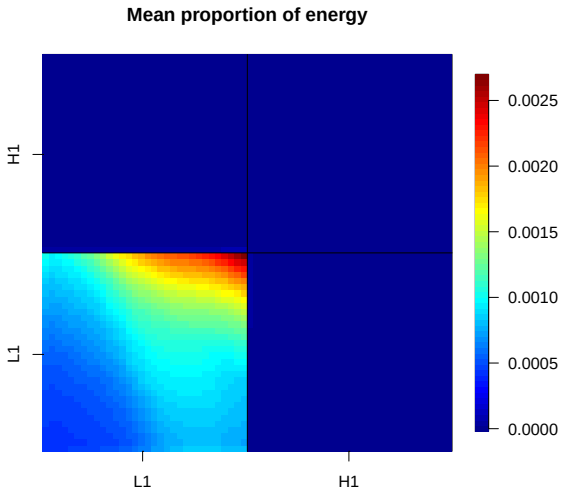

Figure 8: Mean proportion of energy of wavelet coefficients, with the depth of decomposition at 1. "L1" corresponds to scale function on x-axis (horizontal) or y-axis (vertical). "H1" corresponds to wavelet function on x-axis (horizontal) or y-axis (vertical). Square at the bottom left of the figure corresponds to coefficients associated with the 2D scale function. Squares at the top left, the top right and the figures' bottom right correspond to coefficients, respectively associated with vertical, diagonal, and horizontal wavelet (translation direction).

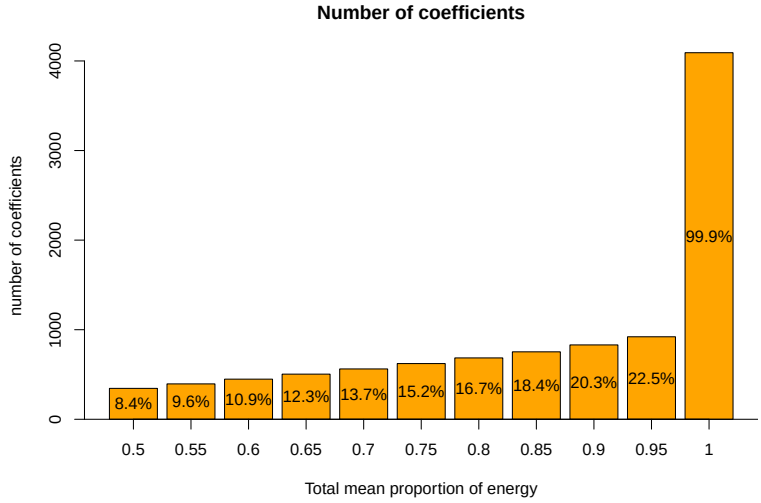

Figure 9: Number of coefficients, which are used on PCA, according to the total mean proportion of energy. The given percentages correspond to the percentage of selected coefficients.

4 Prediction

4.1 Fitting Gaussian Process Regression models

In GpOutput2D, two functions, called km_Fpca2d and gp_Fpca2d, fit Gaussian process (GP) regression model (also called kriging model) on each principal component, which are obtained by Fpca2d. km_Fpca2d uses the km function from DiceKriging package. gp_Fpca2d uses the gp function from kergp package. Both packages allow to create GP models.

kergp is a Laboratory package which performs GP interpolation with an emphasis on user-defined covariance kernels. Furthermore, categorical variables can also be treated as model inputs. However, kergp may evolve in the future. Users interested in stable software for the Analysis of Computer Experiments can used instead DiceKriging package.

km_Fpca2d

```
> # design of experiment in data.frame
> colnames(doe_learn)<-paste("x",1:8,sep="")
> doe_learn<-data.frame(doe_learn)
> # using wavelet basis
> mw <- km_Fpca2d(design=doe_learn,response=FPCA_dwt,control=list(trace=FALSE))
> # using othonormalized B-splines basis
> mB <- km_Fpca2d(design=doe_learn,response=FPCA_bs,control=list(trace=FALSE))
```

gp_Fpca2d

For the example, we use the same scale and variance parameters as estimated in km_Fpca2d. In order to assign scale and variance parameters on each principal component kriging model, a list of kernel function is given in gp_Fpca2d.

```
> library(doFuture)
> #============================
> #   By using wavelet basis
> #============================
>
> ## kernel for each principal component
> myCov <- c()
> for(l in 1:nPC){
+   # model
+   cov_ml <- mw[[l]]@covariance
+
+   #kernel
+   kl<-covTS(inputs = colnames(doe_learn),
+           kernel = "k1Matern5_2",
+           dep = c(range = "input"))
+
+   # allocation of scale and variance parameters
+   coef(kl)<-c(range = cov_ml@range.val, sigma2 = rep(cov_ml@sd2,8))
+
+   myCov <- c(myCov,list(kl) )
+ } # end for l
> ## model by using wavelet basis
>
> gp_w<- gp_Fpca2d(design=doe_learn, response=FPCA_dwt, cov=myCov,estim=FALSE)
> #============================
> #   By using B-splines basis
> #============================
>
> ## kernel for each principal component
> myCov <- c()
> for(l in 1:nPC){
+   # model
+   cov_ml <- mB[[l]]@covariance
+
+   #kernel
+   kl<-covTS(inputs = colnames(doe_learn),
+           kernel = "k1Matern5_2",
+           dep = c(range = "input"))
+
+   # allocation of scale and variance parameters
+   coef(kl)<-c(range = cov_ml@range.val, sigma2 = rep(cov_ml@sd2,8))
+
+   myCov <- c(myCov,list(kl) )
+ } # end for l
```

```
> ## model by using B-splines basis
> gp_B<- gp_Fpca2d(design=doe_learn, response=FPCA_bs, cov=myCov,estim=FALSE)
```

4.2 Prediction

To analyse prediction accuracy, a test sample of size $n_{test} = 1000$ has been built with inputs randomly chosen according to a uniform distribution $\mathcal{U}([-1, 5]^8)$.

```
> ntest<-1000
> # inputs
> NewX <- matrix(runif(ntest*8,min=-1,max=5),ncol=8)
> # outputs
> Ytest <- Campbell2D(NewX,z,z)
> # inputs in data.frame
> colnames(NewX)<-colnames(doe_learn)
> NewX <-data.frame(NewX)
```

A S3 method predict has been developed for km_Fpca2d and gp_Fpca2D. By specifying compute = FALSE, only the kriging mean is computed. If TRUE, the kriging variance (here, the standard deviation is returned) and confidence intervals are computed too. Here, compute is TRUE only for the model built by km_Fpca2d with wavelet basis.

```
> #============================
> #   By using wavelet basis
> #============================
>
> # By using km_Fpca2d
> pw_km <- predict(mw,newdata=NewX,type="UK")
> # By using gp_Fpca2d
> pw_gp <- predict(gp_w,newdata=NewX,type="UK", compute = FALSE)
>
> #============================
> #   By using B-splines basis
> #============================
>
> # By using km_Fpca2d
> pB_km <- predict(mB,newdata=NewX,type="UK", compute = FALSE)
> # By using gp_Fpca2d
> pB_gp <- predict(gp_B,newdata=NewX,type="UK", compute = FALSE)
```

4.3 An example of prediction

```
> i = 20 # example simulation
> par(mfrow=c(1,3))
> # z-axis limit
> zlims <- c(7.9,27.5)
> # number of colors
> nlevel=20
> # real matrix
> rst[]<-rot90(Ytest[,,i])
```

```
> plot(rst,col=rainbow(nlevel),zlim=zlims,legend.width=2,
+      main="Real map")
> # prediction
> rst[]<-rot90(pw_km$mean[,,i])
> plot(rst,col=rainbow(nlevel),zlim=zlims,legend.width=2,
+      main="Estimated map")
> # kriging variance
> library(RColorBrewer)
> color <-colorRampPalette(c("white","yellow","orange","red","darkred","black"))
> rst[]<-rot90(pw_km$sd[,,i]**2)
> plot(rst,col=color(nlevel),legend.width=2,
+      main="kriging variance")
> # 95% confidence interval
> par(mfrow=c(1,2))
> ## lower bound
> rst[]<-rot90(pw_km$lower95[,,i])
> plot(rst,col=rainbow(nlevel),zlim=zlims,legend.width=2,
+      main="95% confidence interval (lower bound)")
> ## upper bound
> rst[]<-rot90(pw_km$upper95[,,i])
> plot(rst,col=rainbow(nlevel),zlim=zlims, legend.width=2,
+      main="95% confidence interval (upper bound)")
>
```

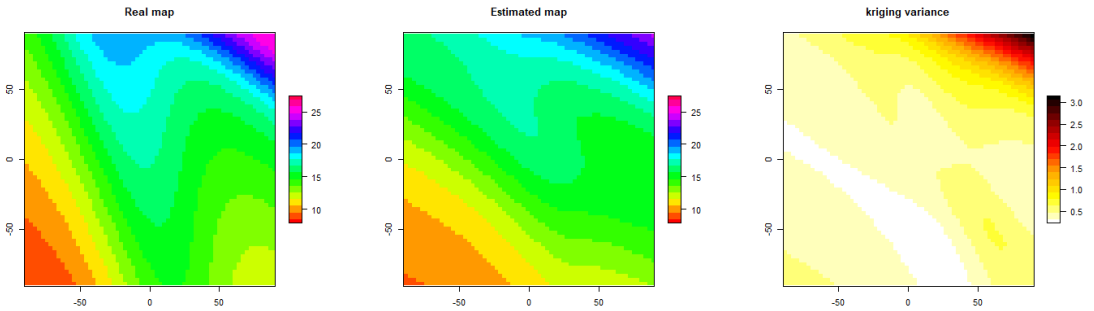

Figure 10: Example of estimated matrix. From left to right : the real matrix, the estimated matrices (which is obtained by km_Fpca2d with wavelet basis), and the prediction variance (it can be assimilated to the kriging variance).

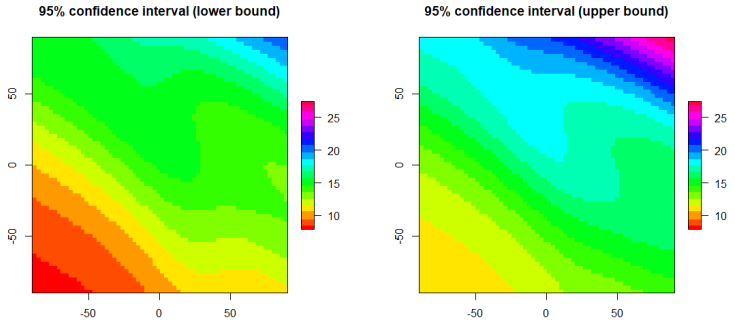

Figure 11: From left to right, lower and upper bound of the 95% confidence interval of the estimated matrix (see Figure 10)

4.4 Prediction accuracy

The function error.predict measures the prediction accuracy by computing the spatial RMSE and Q^2 as follows :

$$\text{RMSE}(\mathbf{z}) = \sqrt{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} \left(y_i(\mathbf{z}) - \hat{y}_i(\mathbf{z})\right)^2}$$

$$Q^2(\mathbf{z}) = 1 - \frac{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i(\mathbf{z}) - \hat{y}_i(\mathbf{z}))^2}{\frac{1}{n_{test}} \sum_{i=1}^{n_{test}} (y_i(\mathbf{z}) - \overline{y}(\mathbf{z}))^2}$$

with $\hat{y}_i(.)$, the estimation of $y_i(.)$, and $\overline{y}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^{n} y_i(\mathbf{z})$. By specifying rtx.scores=FALSE, the prediction accuracy of each principal component kriging model is also measured by RMSE and Q^2. The rtx.scores default is FALSE. To get RMSE and Q^2, we must give the real matrices, the predictions, and the Fpca2d object used to build the kriging models. For the example, error.predict is run on the prediction obtained with km_Fpca2d, for wavelet and B-splines basis. Figure 12 shows the RMSE and Q^2 matrices of both metamodels.

```
> #=========
> # wavelet
> #=========
> err_pw <- error.predict(Ytest,pw_km,FPCA_dwt,rtx.scores=TRUE)
> # Prediction accuracy of score estimations
> print(err_pw$scores$rmse) # RMSE

      PC1       PC2       PC3       PC4       PC5
19.970950 15.373994 11.868038  8.427110  9.718261

> print(err_pw$scores$Q2) # Q2

      PC1       PC2       PC3       PC4       PC5
0.9944059 0.9779924 0.9478785 0.9552067 0.9454740

> #============
> # B-splines
> #============
```

```
> err_pB <- error.predict(Ytest,pB_km,FPCA_bs,rtx.scores=TRUE)
> # Prediction accuracy of score estimations
> print(err_pB$scores$rmse) # RMSE

     PC1      PC2      PC3      PC4      PC5
19.97097 15.37399 11.86802  8.42704  9.71815

> print(err_pB$scores$Q2) # Q2

      PC1       PC2       PC3       PC4       PC5
0.9944059 0.9779924 0.9478786 0.9552073 0.9454752

> #==================
> # RMSE and Q2 matrices
> #==================
>
> # RMSE
> par(mfrow=c(1,2))
> zlims=c(0.31,2.38)
> ## Fpca-wavelet
> rst[]<-rot90(err_pw$y$rmse)
> plot(rst, zlim=zlims, col=rainbow(100),
+      main="RMSE with wavelet basis",legend.width=2)
> ## Fpca-Bsplines
> rst[]<-rot90(err_pB$y$rmse)
> plot(rst, zlim=zlims,col=rainbow(100),
+       main="RMSE with B-splines basis",legend.width=2)
> # Q2
> par(mfrow=c(1,2))
> zlims=c(0.83,0.99)
> color<-colorRampPalette(c("orange","yellow","green","darkgreen","black"))
> ## Fpca-wavelet
> rst[]<-rot90(err_pw$y$Q2)
> plot(rst, zlim=zlims, col=color(100),
+      main="Q2 with wavelet basis",legend.width=1.5)
> ## Fpca-Bsplines
> rst[]<-rot90(err_pB$y$Q2)
> plot(rst, zlim=zlims,col=color(100),
+       main="Q2 with B-splines basis",legend.width=1.5)
>
```

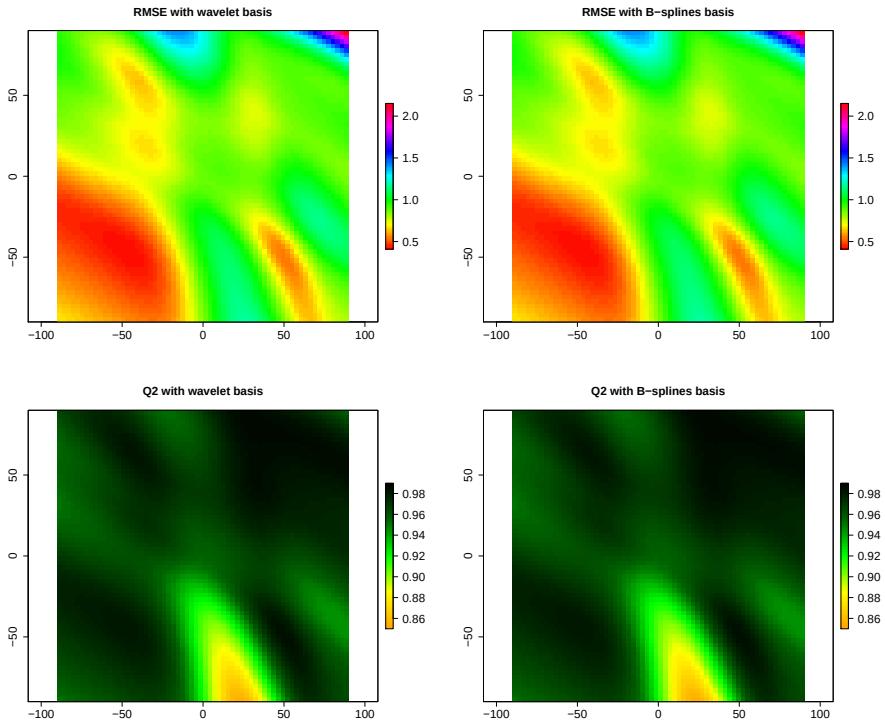

Figure 12: From left to right, the results for FPCA based on wavelets and for FPCA based on B-splines. The top figures are the RMSE matrices. The bottom figures are the Q^2 matrices.

References

Björck, Å. (1994). Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316.

Dupuy, D., Helbert, C., Franco, J., et al. (2015). Dicedesign and diceeval: Two r packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11):1–38.

Liu, X., Nassar, H., and Podgórski, K. (2019). The ob-splines–efficient orthonormalization of the b-splines.

Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

Marrel, A., Iooss, B., Jullien, M., Laurent, B., and Volkova, E. (2010). Global sensitivity analysis for models with spatialy dependent outputs. *Environmetrics*, 22(3):383–397.

Perrin, T., Roustant, O., Rohmer, J., Alata, O., Naulin, J., Idier, D., Pedreros, R., Moncoulon, D., and Tinard, P. (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliability Engineering & System Safety*, 211:107522.

Qin, K. (2000). General matrix representations for b-splines. *The Visual Computer*, 16(3-4):177–186.

Ramsay, J. O. (2006). *Functional data analysis*. Wiley Online Library.

Ramsay, J. O. and Silverman, B. W. (2007). *Applied functional data analysis: methods and case studies*. Springer.

Redd, A. (2012). A comment on the orthogonalization of b-spline basis functions and their derivatives. *Statistics and Computing*, 22(1):251–257.

Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. (2020). Group kernels for gaussian process metamodels with categorical inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):775–806.

Vetterli, M. and Kovacevic, J. (1995). *Wavelets and subband coding*. Number BOOK. Prentice-hall.

Williams, C. K. I. and Rasmussen, C. E. (2006). *Gaussian processes for machine learning*. MIT Press Cambridge, MA.

Cinquième partie

Conclusion et perspectives

Chapitre 7

Conclusion et perspectives

7.1 Contributions de la thèse

Dans la thèse, des méthodologies de méta-modélisation et d'analyse de sensibilité ont été introduites pour les modèles avec des sorties spatiales à grande dimension incluant de fortes discontinuités. Ce travail a été motivé par l'analyse de sensibilité de modèle de submersion marine.

Dans ce but, on propose de combiner le méta-modèle avec l'analyse en composantes principales fonctionnelle (ACPF) pour réduire la dimension de la sortie spatiale, afin de combiner les avantages de l'approximation sur base fonctionnelle et la réduction de dimension par ACP. Pour réduire davantage la dimension, on ajoute une étape de sélection préliminaire. Cette sélection peut être faite à la fois directement sur la base avec une approche de régression pénalisée, ou avec un critère d'énergie après orthonormalisation. Cette seconde approche montre de meilleures performances dans nos expérimentations, à la fois en termes de précision et de temps de calcul, et présente aussi des avantages en termes d'interprétation physique.

Deux types de bases ont été comparées : les ondelettes et les B-splines. Premièrement, la méthodologie a été testée sur un cas test analytique où l'ACPF donne les mêmes résultats que l'approche ACP. Cela montre qu'il n'y a pas de perte de précision quand deux décompositions sont emboîtées dans l'ACP, même en utilisant un nombre restreint de coefficients d'ondelettes bien choisis. Pour le BRGM, nos résultats expérimentaux montrent que la méta-modélisation ACPF est plus précise que l'ACP dans les zones où des irrégularités marquées sont présentes. Pour la CCR, seule l'approche ACPF sur base d'ondelettes est plus précise que l'ACP. Les cartes de submersion marine utilisées dans la thèse sont des matrices de dimension 256×256 : cela permet d'appliquer l'ACP et de comparer les résultats avec l'ACPF. En pratique, des dimensions plus grandes (où l'ACP est difficilement applicable) peuvent être considérées avec notre approche, même si les B-splines sont utilisées. En outre, une analyse de sensibilité est faite spatialement en analysant la structure des composantes principales générés par l'ACPF, et en

calculant les indices de Sobol sur chacune d'entre elles. Cela permet d'interpréter spatialement l'influence des paramètres d'entrée sur la sortie, avec un nombre limité de cartes. L'AS est aussi résumée en un indice scalaire, résumant les résultats obtenus spatialement. Dans la thèse, on propose une extension de l'indice proposé dans [Lamboni et al., 2011], qui est valide pour une base de fonctions quelconque, et plus seulement orthonormée. L'application sur le cas réel de ces indices permet d'identifier les entrées en accord avec les processus de débordement dans la zone étudiée.

Ces développements ont été appliqués à deux simulateurs différents (BRGM et CCR), ainsi qu'à deux cas d'application : une tempête synthétique (voir la section 5.1) et une tempête spécifique, celle de Xynthia (voir la section 5.5). Ces applications ont permis de montrer que la méthode est transposable. Dans le cas de la tempête synthétique, les résultats montrent une forte influence de la marée pour les modèles d'aléa du BRGM et de la CCR (voir la figure 5.22). Dans le cas de la tempête Xynthia, l'importance du simulateur utilisé pour estimer le forçage marin est mis en avant pour les deux modèles. Pour le BRGM, l'analyse de l'ouverture des connexions hydrauliques a montré que la connexion numéros 5 parmi les 7 étudiées a aussi une forte influence, supérieure à celle de la rugosité du terrain qui est presque nulle selon l'indice de sensibilité généralisé (voir la figure 5.26). À l'inverse, pour la CCR, bien que faible selon l'indice généralisé (voir la figure 5.29), l'influence du coefficient de rugosité frott.A (zone urbaine) est à noter (voir la figure 5.28). Les développements ont fait l'objet d'un article publié dans la revue *Reliability Engineering and System Safety*, et d'un package **R** (voir le chapitre 6).

7.2 Pistes d'amélioration et de réflexion

Plusieurs axes d'amélioration ont été identifiés. Premièrement, la prédiction de cartes lorsqu'une inondation se produit et celles lorsqu'il y en a aucune est encore difficile, bien que la profondeur d'eau prédite soit faible en l'absence d'inondations. Cela est lié à certains effets de seuil qui contrôlent les processus côtiers. Si le niveau d'eau à la côte (qui résulte des caractéristiques des ondes, des tempêtes et des marées) est inférieur à un certain seuil, les inondations ne peuvent pas se produire : les hauteurs d'eau à terre restent donc nulles. Autrement dit, des inondations peuvent survenir et une partie des terres peut être inondée, à condition que le niveau d'eau à la côte augmente légèrement et dépasse un certain seuil. Pour résoudre cette difficulté, les pistes suivantes peuvent être explorées : les méthodes de classification afin d'apprendre les entrées où aucune inondation ne se produit (voir un exemple dans [Rohmer et al., 2018] qui propose une approche basée sur les forêts aléatoires), ou en ajoutant des contraintes aux modèles de krigeage [Lopera, 2019] sur chaque composante principale. Pour la dernière solution proposée, il faudrait détecter les scores de l'ACP associés aux cartes où aucune inondation ne se produit.

Deuxièmement, bien que l'utilisation d'une base fonctionnelle vise à préserver la

régularité spatiale, certaines zones inondées, en gris dans les figures 5.18 et 5.19, ne sont pas suffisamment connectées ensemble dans les cartes prédites et par conséquent connectées à la mer. Cependant, dans le modèle physique, la propagation du flux provient de la mer, et les zones inondées sont toujours continues, sauf si le modèle représente des connexions hydrauliques, telles que des buses. Une solution possible à ce problème serait d'ajouter un critère de régularité au critère de l'énergie, utilisé pour sélectionner les coefficients de la base. Dans ce but, l'alternative du critère Lasso, présentée dans la section 3.2.1.2, pourrait être approfondie, afin de le rendre applicable au cas de cartes à grandes dimensions.

Troisièmement, les indices de sensibilité ont été estimés en utilisant la variance comme mesure d'incertitude. Cette dernière pourrait ne pas être adaptée pour représenter les phénomènes physiques comportant des effets de seuil (ce qui peut induire certaines multi-modalités dans la distribution de probabilité de la sortie), comme cela peut être le cas des submersions marines. Des travaux futurs devraient considérer des mesures d'incertitudes alternatives (comme les mesures de dépendances [Da Veiga, 2015, De Lozzo and Marrel, 2017]).

Quatrièmement, dans les sections 5.4.1 et 5.5.2, une interprétation spatiale de l'analyse de sensibilité a été faite en analysant la structure spatiale des composantes principales (ou fonctions propres), générées à partir de l'échantillon d'apprentissage du méta-modèle. Les observations utilisées pour estimer les indices de sensibilité sont des estimations obtenues à partir de ce méta-modèle. Cependant, les composantes principales ne prennent pas en compte l'incertitude de ces prédictions. Pour des entrées inconnues, le méta-modèle devrait donc prédire les scores de l'ACP (ou ACPF) tout en réadaptant les composantes principales à la connaissance de l'incertitude de ces prédictions. Les processus gaussiens étant utilisés dans la thèse, le problème revient donc à déterminer la loi de distribution des composantes principales. Des formulations probabilistes de l'ACP proposées par [Tipping and Bishop, 1999, Bishop, 1999] pourraient être exploitées.

La section 5.6 présente une cinquième perspective : développer une méthode qui associerait les simulations des modèles d'aléa du BRGM et de la CCR, afin d'améliorer les estimations des inondations. Une méthode testée est de concaténer les simulations faites par les deux modèles d'aléa, et d'ajouter une variable catégorielle en entrée du méta-modèle qui indique le modèle utilisé. Pour les deux modèles, les résultats (voir la figure 5.30) ont montré qu'en les associant la probabilité de détecter un sinistre (l'inondation d'un bien est à la fois estimée et observée en réalité) augmente par rapport aux estimations initiales du BRGM et de la CCR. Cependant, la probabilité de fausse détection (un bien est estimé inondé alors qu'il ne l'a pas été en réalité) augmente aussi, mais reste faible. Des travaux complémentaires dans le futur sont ici nécessaires et devraient permettre de confirmer le réel intérêt de combiner des simulateurs numériques d'aléa de natures complètement différentes.

Bibliographie

[Bachoc, 2013] Bachoc, F. (2013). Cross validation and maximum likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. *Computational Statistics & Data Analysis*, 66 :55–69.

[Bates et al., 2005] Bates, P. D., Dawson, R. J., Hall, J. W., Horritt, M. S., Nicholls, R. J., Wicks, J., and Hassan, M. A. A. M. (2005). Simplified two-dimensional numerical modelling of coastal flooding and example applications. *Coastal Engineering*, 52(9) :793–810.

[Bates et al., 2010] Bates, P. D., Horritt, M. S., and Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1-2) :33–45.

[Bishop, 1999] Bishop, C. M. (1999). Bayesian PCA. In *Advances in neural information processing systems*, pages 382–388.

[Björck, 1994] Björck, Å. (1994). Numerics of Gram-Schmidt orthogonalization. *Linear Algebra and Its Applications*, 197 :297–316.

[Bratley and Fox, 1988] Bratley, P. and Fox, B. L. (1988). Algorithm 659 : Implementing Sobol's quasirandom sequence generator. *ACM Transactions on Mathematical Software (TOMS)*, 14(1) :88–100.

[Campbell et al., 2006] Campbell, K., McKay, M. D., and Williams, B. J. (2006). Sensitivity analysis when model outputs are functions. *Reliability Engineering & System Safety*, 91(10) :1468–1472.

[Chen et al., 2011] Chen, T., Hadinoto, K., Yan, W., and Ma, Y. (2011). Efficient meta-modelling of complex process simulations with time–space-dependent outputs. *Computers & chemical engineering*, 35(3) :502–509.

[Chiles and Delfiner, 1999] Chiles, J. and Delfiner, A. (1999). Geostatistics : Modelling spatial uncertainty : Wiley interscience. *New York*.

[Chiles and Delfiner, 2012] Chiles, J.-P. and Delfiner, P. (2012). *Geostatistics : modeling spatial uncertainty, 2nd Edition*. Wiley series in probability and statistics.

[Cressie, 1993] Cressie, N. A. (1993). Statistics for spatial data. *John Wiley and Sons Inc., New York*.

[Cukier et al., 1978] Cukier, R., Levine, H., and Shuler, K. (1978). Nonlinear sensitivity analysis of multiparameter model systems. *Journal of computational physics*, 26(1) :1–42.

[Da Veiga, 2015] Da Veiga, S. (2015). Global sensitivity analysis with dependence measures. *Journal of Statistical Computation and Simulation*, 85(7) :1283–1305.

[Daubechies, 1988] Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Communications on pure and applied mathematics*, 41(7) :909–996.

[De Lozzo and Marrel, 2017] De Lozzo, M. and Marrel, A. (2017). Sensitivity analysis with dependence and variance-based measures for spatio-temporal numerical simulators. *Stochastic environmental research and risk assessment*, 31(6) :1437–1453.

[Dupuy et al., 2015] Dupuy, D., Helbert, C., Franco, J., et al. (2015). DiceDesign and DiceEval : Two R packages for design and analysis of computer experiments. *Journal of Statistical Software*, 65(11) :1–38.

[Efron and Tibshirani, 1994] Efron, B. and Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC press.

[Faivre et al., 2016] Faivre, R., Iooss, B., Mahévas, S., Makowski, D., and Monod, H. (2016). *Analyse de sensibilité et exploration de modèles : application aux sciences de la nature et de l'environnement*. Editions Quae.

[FFSA, 2011] FFSA, G. (2011). La tempête Xynthia du 28 février 2010. Bilan chiffré au 31 décembre 2010. Technical report, Association française de l'assurance.

[Gamboa et al., 2020] Gamboa, F., Gremaud, P., Klein, T., and Lagnoux, A. (2020). Global Sensitivity Analysis : a new generation of mighty estimators based on rank statistics. *arXiv preprint arXiv :2003.01772*.

[Garry et al., 1999] Garry, G., Edmond, G., and Levoy, F. (1999). Plans de prévention des risques littoraux (ppr) : Guide méthodologique. *Direction de la précention des pollutions et des risques et Direction de l'aménagement foncier et de l'urbanisme*.

[Gençay et al., 2001] Gençay, R., Selçuk, F., and Whitcher, B. J. (2001). *An introduction to wavelets and other filtering methods in finance and economics*. Elsevier.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning : data mining, inference, and prediction*. Springer Science & Business Media.

[Homma and Saltelli, 1996] Homma, T. and Saltelli, A. (1996). Importance measures in global sensitivity analysis of nonlinear models. *Reliability Engineering & System Safety*, 52(1) :1–17.

[Iooss, 2011] Iooss, B. (2011). Revue sur l'analyse de sensibilité globale de modèles numériques. *Journal de la Société Française de Statistique*, 152(1) :1–23.

[Iooss and Lemaître, 2015] Iooss, B. and Lemaître, P. (2015). A review on global sensitivity analysis methods in : Uncertainty management in simulation-optimization of complex systems. In *Operations research/computer science interfaces series 59*, pages 101–122. Springer US.

[Iooss and Ribatet, 2009] Iooss, B. and Ribatet, M. (2009). Global sensitivity analysis of computer models with functional inputs. *Reliability Engineering & System Safety*, 94(7) :1194–1204.

[Jia and Taflanidis, 2013] Jia, G. and Taflanidis, A. A. (2013). Kriging metamodeling for approximation of high-dimensional wave and surge responses in real-time storm/hurricane risk assessment. *Computer Methods in Applied Mechanics and Engineering*, 261 :24–38.

[Jolliffe, 2002] Jolliffe, I. (2002). Principal Component Analysis, 2nd edn. Series : Springer Series in Statistics, XXIX, 487. *illus. Springer, NY*, page 28.

[Kennedy et al., 2000] Kennedy, M., Kopp, S., et al. (2000). *Understanding map projections*. Esri Redlands, CA.

[Khuri and Good, 1989] Khuri, A. I. and Good, I. (1989). The parameterization of orthogonal matrices : A review mainly for statisticians. *South African Statistical Journal*, 23(2) :231–250.

[Krige, 1951] Krige, D. G. (1951). A statistical approach to some basic mine valuation problems on the Witwatersrand. *Journal of the Southern African Institute of Mining and Metallurgy*, 52(6) :119–139.

[Lamboni et al., 2011] Lamboni, M., Monod, H., and Makowski, D. (2011). Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliability Engineering & System Safety*, 96(4) :450–459.

[Lazure and Dumas, 2008] Lazure, P. and Dumas, F. (2008). An external–internal mode coupling for a 3D hydrodynamical model for applications at regional scale (MARS). *Advances in water resources*, 31(2) :233–250.

[Li et al., 2020] Li, M., Wang, R.-Q., and Jia, G. (2020). Efficient dimension reduction and surrogate-based sensitivity analysis for expensive models with high-dimensional outputs. *Reliability Engineering & System Safety*, 195 :106725.

[Lilburne and Tarantola, 2009] Lilburne, L. and Tarantola, S. (2009). Sensitivity analysis of spatial models. *International Journal of Geographical Information Science*, 23(2) :151–168.

[Liu et al., 2019] Liu, X., Nassar, H., and PodgÓrski, K. (2019). Splinets–efficient orthonormalization of the B-splines. *arXiv preprint arXiv :1910.07341*.

[Lopera, 2019] Lopera, A. F. L. (2019). *Gaussian Process Modelling under Inequality Constraints*. PhD thesis, Université de Lyon.

[López-Lopera et al., 2020] López-Lopera, A. F., Idier, D., Rohmer, J., and Bachoc, F. (2020). Multi-Output Gaussian Processes with Functional Data : A Study on Coastal Flood Hazard Assessment. *arXiv preprint arXiv :2007.14052*.

[Ma et al., 2019] Ma, P., Mondal, A., Konomi, B., Hobbs, J., Song, J., and Kang, E. (2019). Computer Model Emulation with High-Dimensional Functional Output in Large-Scale Observing System Uncertainty Experiments. *arXiv preprint arXiv :1911.09274*.

[Mallat, 1999] Mallat, S. (1999). *A wavelet tour of signal processing*. Elsevier.

[Mallat, 1989] Mallat, S. G. (1989). A theory for multiresolution signal decomposition : the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, 11(7) :674–693.

[Marrel et al., 2011] Marrel, A., Iooss, B., Jullien, M., Laurent, B., and Volkova, E. (2011). Global sensitivity analysis for models with spatially dependent outputs. *Environmetrics*, 22(3) :383–397.

[Marrel et al., 2009] Marrel, A., Iooss, B., Laurent, B., and Roustant, O. (2009). Calculations of Sobol indices for the Gaussian process metamodel. *Reliability Engineering & System Safety*, 94(3) :742–751.

[Marrel et al., 2015] Marrel, A., Perot, N., and Mottet, C. (2015). Development of a surrogate model and sensitivity analysis for spatio-temporal numerical simulators. *Stochastic environmental research and risk assessment*, 29(3) :959–974.

[Naulin et al., 2015] Naulin, J., Moncoulon, D., Le Roy, S., Pedreros, R., Idier, D., and Oliveros, C. (2015). Estimation of insurance related losses resulting from coastal flooding in france. *Natural Hazards and Earth System Sciences Discussions*, 3(4) :2811–2846.

[Perrin et al., 2021] Perrin, T., Roustant, O., Rohmer, J., Alata, O., Naulin, J., Idier, D., Pedreros, R., Moncoulon, D., and Tinard, P. (2021). Functional principal component analysis for global sensitivity analysis of model with spatial output. *Reliability Engineering & System Safety*, 211 :107522.

[Pinheiro and Bates, 2009] Pinheiro, J. and Bates, D. (2009). *Mixed-effects models in S and S-PLUS*. Statistics and Computing. Springer New York.

[Pinheiro and Bates, 1996] Pinheiro, J. C. and Bates, D. M. (1996). Unconstrained parametrizations for variance-covariance matrices. *Statistics and computing*, 6(3) :289–296.

[Qin, 2000] Qin, K. (2000). General matrix representations for B-splines. *The Visual Computer*, 16(3-4) :177–186.

[R Core Team, 2020] R Core Team (2020). *R : A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[Ramsay and Silverman, 2004] Ramsay, J. and Silverman, B. (2004). Functional data analysis. *Encyclopedia of Statistical Sciences*, 4.

[Rasmussen and Williams, 2006] Rasmussen, C. E. and Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

[Redd, 2012] Redd, A. (2012). A comment on the orthogonalization of B-spline basis functions and their derivatives. *Statistics and Computing*, 22(1) :251–257.

[Rohmer et al., 2018] Rohmer, J., Idier, D., Paris, F., Pedreros, R., and Louisor, J. (2018). Casting light on forcing and breaching scenarios that lead to marine inundation : Combining numerical simulations with a random-forest classification approach. *Environmental Modelling & Software*, 104 :64–80.

[Roustant et al., 2020] Roustant, O., Padonou, E., Deville, Y., Clément, A., Perrin, G., Giorla, J., and Wynn, H. (2020). Group kernels for Gaussian process metamodels with categorical inputs. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2) :775–806.

[Saltelli, 2002] Saltelli, A. (2002). Making best use of model evaluations to compute sensitivity indices. *Computer physics communications*, 145(2) :280–297.

[Saltelli et al., 2008] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. (2008). *Global sensitivity analysis : the primer*. John Wiley & Sons.

[Schaefer, 1990] Schaefer, J. T. (1990). The critical success index as an indicator of warning skill. *Weather and forecasting*, 5(4) :570–575.

[Shepard et al., 2015] Shepard, R., Brozell, S. R., and Gidofalvi, G. (2015). The representation and parametrization of orthogonal matrices. *The Journal of Physical Chemistry A*, 119(28) :7924–7939.

[Sobol, 1993] Sobol, I. M. (1993). Sensitivity estimates for nonlinear mathematical models. *Mathematical modelling and computational experiments*, 1(4) :407–414.

[Tibshirani, 1996] Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society : Series B (Methodological)*, 58(1) :267–288.

[Tipping and Bishop, 1999] Tipping, M. E. and Bishop, C. M. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)*, 61(3) :611–622.

[Tissot and Prieur, 2012] Tissot, J.-Y. and Prieur, C. (2012). Bias correction for the estimation of sensitivity indices based on random balance designs. *Reliability Engineering & System Safety*, 107 :205–213.

[Wackernagel, 2013] Wackernagel, H. (2013). *Multivariate geostatistics : an introduction with applications*. Springer Science & Business Media.

[Xiao and Li, 2016] Xiao, H. and Li, L. (2016). Discussion of paper by matieyendou lamboni, hervé monod, david makowski "multivariate sensitivity analysis to measure global contribution of input factors in dynamic models", reliab. eng. syst. saf. 99 (2011) 450–459. *Reliability Engineering & System Safety*, 147 :194–195.

[Zhang and Notz, 2015] Zhang, Y. and Notz, W. I. (2015). Computer experiments with qualitative and quantitative variables : A review and reexamination. *Quality Engineering*, 27(1) :2–13.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT :

Tran Vi-vi Élodie PERRIN

Metamodelling and sensitivity analysis for models with spatial output. Application to coastal flooding models.

Speciality : Applied Mathematics

Keywords : Global sensitivity analysis, Gaussian processes, spatial data, functional principal component analysis, wavelets, B-splines

Abstract :

Motivated by the risk assessment of coastal flooding, the numerical hydrodynamic models of the BRGM and the CCR are considered. Their outputs are flood maps. The aim is to perform a sensitivity analysis (SA) to quantify and hierarchize the influence of the input parameters on the output. The application of functional PCA (FPCA) is proposed to reduce both computation time and spatial output dimension. The output is decomposed on a basis of functions designed to handle local variations, such as wavelets or B-splines. PCA with an ad-hoc metric is applied on the most important coefficients, according to an energy criterion after basis orthonormalization, or on the initial basis with a penalized regression approach. Fast-to-evaluate metamodels (such as Kriging) are built on the first principal components, on which SA can be done. As a by-product, we obtain analytical formulas for variance-based sensitivity indices, generalizing a known formula assuming the orthonormality of basis functions. The whole methodology is applied to an analytical case and two coastal flooding cases. Gains in accuracy and computation time have been obtained. An R package has been developed, which allows sharing the research outputs.

École Nationale Supérieure des Mines
de Saint-Étienne

NNT :

Tran Vi-vi Élodie PERRIN

Méta-modélisation et analyse de sensibilité pour les modèles avec sortie spatiale. Application aux modèles de submersion marine.

Spécialité : Mathématiques Appliquées

Mots clefs : Analyse de sensibilité globale, processus gaussien, données spatiales, analyse en composantes principales fonctionnelle, ondelettes, B-splines

Résumé :

Cette thèse est motivée par l'évaluation des risques de submersions marines. On considère les modèles hydrodynamiques numériques développés par le BRGM et la CCR. La sortie de ces simulateurs est une carte d'inondation. L'objectif est de réaliser une analyse de sensibilité (AS) afin de mesurer et de hiérarchiser l'influence des paramètres d'entrée sur la sortie. Afin de réduire le temps de calcul des modèles et la dimension de la sortie spatiale, on propose d'utiliser l'ACP fonctionnelle (ACPF). La sortie est décomposée dans une base de fonctions, adaptée pour traiter les variations locales, telle que les ondelettes ou les B-splines. Une ACP avec une métrique ad-hoc est appliquée aux coefficients les plus importants, selon un critère d'énergie après orthonormalisation de la base, ou directement sur la base originale avec une approche de régression pénalisée. Des méta-modèles (comme le krigeage) sont construits sur les premières composantes principales, sur lesquels peut être réalisée l'AS. Comme résultat complémentaire, une formule analytique est obtenue pour les indices de sensibilité basés sur la variance, généralisant celle connue pour des bases orthonormées. L'ensemble des travaux a été appliqué à un cas analytique et deux cas de submersion marine, sur lesquels des gains en précision et en temps de calculs ont été obtenus. Un package R a été développé permettant la diffusion des travaux réalisés.