



Échantillonnages Monte Carlo et quasi-Monte Carlo pour l'estimation des indices de Sobol' : application à un modèle transport-urbanisme

Laurent Gilquin

► To cite this version:

Laurent Gilquin. Échantillonnages Monte Carlo et quasi-Monte Carlo pour l'estimation des indices de Sobol' : application à un modèle transport-urbanisme. Analyse numérique [math.NA]. Université Grenoble Alpes, 2016. Français. $\langle$ NNT : 2016GREAM042 $\rangle$. $\langle$ tel-01403914v4 $\rangle$

HAL Id: tel-01403914

<https://theses.hal.science/tel-01403914v4>

Submitted on 10 Jan 2018

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

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



HAL Authorization

THÈSE

Pour obtenir le grade de

**DOCTEUR DE la Communauté UNIVERSITÉ
GRENOBLE ALPES**

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 Août 2006

Présentée par

Laurent Gilquin

Thèse dirigée par **Clémentine Prieur**
et codirigée par **Elise Arnaud**

préparée au sein **du Laboratoire Jean Kuntzmann**
et de **l'Ecole Doctorale Mathématiques, Sciences et Technologies**
de l'Information, Informatique

Echantillonnages Monte Carlo et quasi-Monte Carlo pour l'estima- tion des indices de Sobol'. Ap- plication à un modèle transport- urbanisme.

Thèse soutenue publiquement le **17/10/2016**,
devant le jury composé de :

Olivier Roustant

Professeur, Ecole des Mines de Saint-Etienne, Président

Art B. Owen

Professeur, Stanford University, Rapporteur

Luc Pronzato

Directeur de recherche, CNRS/I3S, Rapporteur

Nicolas Coulombel

Maître de conférences, Université Paris Est/LVMT, Examineur

Hervé Monod

Directeur de recherche, INRA/MaIAGE, Examineur

Clémentine Prieur

Professeur, Université Grenoble Alpes/LJK, Directrice de thèse

Elise Arnaud

Maître de conférences, Université Grenoble Alpes/LJK, Co-Directrice de thèse



Remerciements

Je tiens tout d'abord à remercier mes deux directrices de thèse Clémentine Prieur et Elise Arnaud d'une part pour m'avoir proposé de travailler sur un sujet intéressant m'ayant permis de découvrir des thématiques qui m'étaient inconnues ; mais aussi pour leur encadrement rigoureux, leur disponibilité et leur soutien au cours de ces trois années.

Je tiens ensuite à remercier les membres de mon jury de thèse dont les questions pertinentes soulevées dans les rapports et lors de la soutenance ont permis d'enrichir ce manuscrit. Je remercie plus particulièrement Hervé Monod pour sa collaboration qui a contribué à bonifier plusieurs de mes travaux.

Je termine en remerciant de manière générale les équipes STEEP et AIRSEA de m'avoir accueilli et au sein desquelles j'ai pu travailler dans une ambiance agréable et constructive. Je remercie plus particulièrement Marie-Anne et Anne pour leur aide dans la majorité des démarches administratives et pour avoir toujours été disponibles et à l'écoute. Je conclus bien évidemment en remerciant les pongistes, culturistes, tennismen, footballeurs, skieurs de ces deux équipes pour les très bons moments de détente partagés.

Résumé : Le développement et l'utilisation de modèles intégrés transport-urbanisme sont devenus une norme pour représenter les interactions entre l'usage des sols et le transport de biens et d'individus sur un territoire. Ces modèles sont souvent utilisés comme outils d'aide à la décision pour des politiques de planification urbaine.

Les modèles transport-urbanisme, et plus généralement les modèles mathématiques, sont pour la majorité conçus à partir de codes numériques complexes. Ces codes impliquent très souvent des paramètres dont l'incertitude est peu connue et peut potentiellement avoir un impact important sur les variables de sortie du modèle.

Les méthodes d'analyse de sensibilité globales sont des outils performants permettant d'étudier l'influence des paramètres d'un modèle sur ses sorties. En particulier, les méthodes basées sur le calcul des indices de sensibilité de Sobol' fournissent la possibilité de quantifier l'influence de chaque paramètre mais également d'identifier l'existence d'interactions entre ces paramètres.

Dans cette thèse, nous privilégions la méthode dite à base de plans d'expériences répliqués encore appelée méthode répliquée. Cette méthode a l'avantage de ne requérir qu'un nombre relativement faible d'évaluations du modèle pour calculer les indices de Sobol' d'ordre un et deux.

Cette thèse se focalise sur des extensions de la méthode répliquée pour faire face à des contraintes issues de notre application sur le modèle transport-urbanisme Tranus, comme la présence de corrélations entre paramètres et la prise en compte de sorties multidimensionnelles.

Nos travaux proposent également des approches récursives pour l'estimation séquentielle des indices de Sobol'. Ces approches offrent une meilleure précision sur l'estimation des indices tout en permettant de recycler des premiers jeux d'évaluations du modèle. Une de ces approches correspond à une version récursive de la méthode répliquée et repose sur la construction itérative d'hypercubes latins et de tableaux orthogonaux stratifiés. Une seconde approche repose sur l'utilisation d'un échantillonnage quasi-Monte Carlo et la construction d'un nouveau critère d'arrêt.

Nous présentons également une application de nos contributions pour le calage du modèle de transport-urbanisme Tranus.

Mots clés : Analyse de sensibilité, indices de Sobol', plans d'expériences, méthode répliquée, estimation récursive, tableaux orthogonaux, quasi-Monte Carlo.

Abstract : Land Use and Transportation Integrated (LUTI) models have become a norm for representing the interactions between land use and the transportation of goods and people in a territory. These models are mainly used to evaluate alternative planning scenarios, simulating their impact on land cover and travel demand.

LUTI models and other mathematical models used in various fields are most of the time based on complex computer codes. These codes often involve poorly-known inputs whose uncertainties can have significant effects on the model outputs.

Global sensitivity analysis methods are useful tools to study the influence of the model inputs on its outputs. Among the large number of available approaches, the variance based method introduced by Sobol' allows to calculate sensitivity indices called Sobol' indices. These indices quantify the influence of each model input on the outputs and can detect existing interactions between inputs.

In this framework, we favor a particular method based on replicated designs of experiments called replication method. This method appears to be the most suitable for our application and is advantageous as it requires a relatively small number of model evaluations to estimate first-order or second-order Sobol' indices.

This thesis focuses on extensions of the replication method to face constraints arising in our application on the LUTI model Tranus, such as the presence of dependency among the model inputs, as far as multivariate outputs.

Aside from that, we propose recursive approaches to sequentially estimate Sobol' indices. With these approaches, more accurate Sobol' estimates are obtained while recycling previous sets of model evaluations. One of these approaches corresponds to a recursive version of the replication method and is based on the iterative construction of stratified designs, latin hypercubes and orthogonal arrays. A second approach combines the use of quasi-Monte Carlo sampling and the construction of a new stopping criterion.

An application of our contributions on the LUTI model Tranus is presented.

Keywords : Sensitivity analysis, Sobol' indices, design of experiments, replication method, recursive estimation, orthogonal arrays, quasi-Monte Carlo.

Table des matières

Liste des travaux	1
Introduction	3
1 Plans d'expériences numériques : un tour d'horizon	9
1.1 Des plans classiques aux plans numériques	10
1.1.1 Principes fondamentaux	10
1.1.2 Plans factoriels complets et fractionnaires	11
1.1.3 Vers les plans d'expériences numériques	12
1.2 Plans stratifiés	13
1.2.1 Grilles régulières	14
1.2.2 Hypercubes latins	15
1.2.3 Tableaux orthogonaux	17
1.2.4 Hypercubes latins basés sur tableaux orthogonaux	20
1.3 Suites digitales et réseaux digitaux	21
1.3.1 (t,m,d)-réseaux et (t,d)-suites	22
1.3.2 Suites de Sobol'	23
1.3.3 Méthodes de randomisation	26
1.4 Critères de qualité	29
1.4.1 Discrépance	30
1.4.2 Distance entre points	31
1.4.3 Arbre de poids minimal	32
1.5 Synthèse	34
2 Estimation des indices de Sobol' : méthodes Monte Carlo	37
2.1 Indices de sensibilité	37
2.1.1 Mesure d'importance	38
2.1.2 Décomposition ANOVA	39
2.1.3 Indices de Sobol'	39
2.2 Introduction aux méthodes d'estimation	41
2.2.1 Méthode de Sobol'	41
2.2.2 Extensions de Saltelli	43
2.2.3 Autres estimateurs	45
2.3 Introduction à la méthode répliquée	47
2.3.1 Plans d'expériences répliqués	47
2.3.2 Méthode répliquée	48
2.3.3 Synthèse	52

3	Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces	55
3.1	Introduction	57
3.2	Definition of Sobol' indices	58
3.2.1	Sobol' indices for independent inputs	58
3.2.2	Sobol' indices for dependent inputs	60
3.2.3	Standard estimation of Sobol' indices	61
3.3	Replication method	61
3.3.1	Replication method for first-order Sobol' indices estimation	61
3.3.2	Replication method for closed second-order Sobol' indices estimation	62
3.4	Sampling strategy under linear ordered constraints	65
3.4.1	Sampling strategy based on the simplex to handle constraints	65
3.4.2	Space filling strategy	67
3.4.3	Equidistribution properties	67
3.5	Summary of the subdivision method	71
3.6	Application to test functions	72
3.6.1	Influence of designs on the replication method	72
3.6.2	Replication method versus classical methods	73
3.7	Conclusion	82
3.A	Illustration of the replication method for the estimation of first-order indices	84
3.B	Illustration of the replication method for the estimation of closed second-order indices	85
3.C	Illustration of Algorithm 2 for $k=2$ and $n=9$	89
4	Recursive estimation procedure of Sobol' indices based on replicated designs	93
4.1	Introduction	94
4.2	Recursive estimation of Sobol' indices	95
4.2.1	Definition of Sobol' indices	95
4.2.2	Estimation of Sobol' indices	97
4.2.3	Replication procedure and associated designs	97
4.2.4	Recursive procedure	100
4.3	Nested space-filling designs	102
4.3.1	Nested Latin hypercube design	102
4.3.2	Nested orthogonal array of strength two	103
4.4	Space-filling properties and application	105
4.4.1	Space-filling properties	105
4.4.2	Application to a toy example	108

5	Reliable error estimation for Sobol' indices	115
5.1	Introduction	116
5.2	Digital sequences background	117
5.2.1	Preliminaries	117
5.2.2	Sobol' sequences	118
5.3	Iterative constructions of replicated point sets	119
5.3.1	Multiplicative approach	119
5.3.2	Additive Approach	120
5.4	Space-filling properties	121
5.5	Introduction	123
5.6	Background on Sobol' indices	124
5.6.1	Definition of Sobol' indices	124
5.6.2	Estimation of Sobol' indices	125
5.6.3	Towards a reliable estimation	127
5.7	Reliable error bound for Sobol' indices	127
5.7.1	Reliable integral estimation using digital sequences	127
5.7.2	Extension to Sobol' indices	129
5.8	Sequential estimation procedure	131
5.8.1	Sequential algorithm and cost	131
5.8.2	Improvement	132
5.9	Applications	134
5.9.1	Classical test functions	134
5.9.2	Real case model	139
5.10	Conclusion	142
6	Sensitivity analysis and optimisation of a Land Use and Transport Integrated model	145
6.1	Introduction	146
6.2	Description of Tranus and problem Statement	147
6.2.1	General structure of the model	147
6.2.2	The land use and activity module	148
6.2.3	Problem statement	150
6.3	Calibration procedure's main tools	152
6.3.1	Global Sensitivity Analysis : replication procedure for multi-variate outputs	152
6.3.2	Stochastic optimisation : EGO algorithm	155
6.4	Application to Tranus	157
6.4.1	Calibration procedure algorithm	157
6.4.2	Analytical optimisation procedure	158
6.4.3	Results	159
6.A	Tranus system of equations	165
6.B	Bootstrap confidence intervals	166

Conclusion et perspectives	169
A Additional notes	171
A.1 Unwanted duplicates	171
A.1.1 Case of first-order indices	171
A.1.2 Case of closed second-order indices	174
A.2 Orthogonal arrays and gain	174
B Notes additionnelles	177
B.1 Présence de collisions	177
B.1.1 Cas des indices d'ordre un	177
B.1.2 Cas des indices fermés d'ordre deux	180
B.2 Tableaux orthogonaux et gain	181
C Code Maple	183
Bibliographie	187

Liste des travaux

Articles dans journaux internationaux avec comité de relecture

- [GAP15], L. Gilquin and E. Arnaud and C. Prieur. Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces, 2015. *Information and Inference : a journal of the IMA* ; volume 10.

Cet article a obtenu le label de reproductibilité.

Soumis

- [GAMP16], L. Gilquin and E. Arnaud and H. Monod and C. Prieur. Recursive estimation procedure of Sobol' indices based on replicated designs, 2016, soumis.
- [GACP16], L. Gilquin and T. Capelle and E. Arnaud and C. Prieur. Sensitivity analysis and optimisation of a Land Use and Transport Integrated model, 2016, soumis.
- [GRA⁺16], L. Gilquin and L. A. Jiménez Rugama and E. Arnaud and F. J. Hickernell and H. Monod and C. Prieur. Iterative construction of replicated designs based on Sobol' sequences, 2016, soumis.
- [GR16], L. Gilquin and L. A. Jiménez Rugama. Reliable error estimation for Sobol' indices, 2016, soumis.

[GAP15] constitue le chapitre 3, [GAMP16] le chapitre 4, [GACP16], [GRA⁺16, GR16] le chapitre 5 et [GRA⁺16] le chapitre 6.

Présentations

Conférences internationales

- *Uncertainty in Computer Models 2014*, Sheffield, 28-30 Juillet 2014. Titre de la présentation : Sobol' indices estimation of constrained inputs based on simplex structure and replicated design. Site web : <http://www.mucm.ac.uk/UCM2014.html>.
- *SIAM Conference on Uncertainty Quantification 2016*, Lausanne, 5-8 Avril 2016. Titre de la présentation : Recursive estimation procedure of Sobol' indices based on replicated designs. Site web : http://meetings.siam.org/session/dsp_programsess.cfm?SESSIONCODE=21949.

Conférences nationales

- *Journées de Statistique 2015 (SFdS)*, Lille, 1-5 Juin 2015. Titre de la présentation : L. Gilquin et E. Arnaud et C. Prieur, Plans emboîtés pour l'estimation itérative des indices de Sobol' par méthode répliquée, http://papersjds15.sfds.asso.fr/submission_48.pdf.
- *Journées MASCOT-NUM 2016*, Toulouse, 23-25 Mars 2016. Titre de la présentation : Replication procedure for the estimation of Sobol' indices and its extensions. Site web : <http://mascot2016.sciencesconf.org/resource/page/id/2>.
- *Journées de Statistique 2016 (SFdS)*, Montpellier, 30 Mai - 3 Juin 2016. Titre de la présentation : L. Gilquin et E. Arnaud et C. Prieur, Analyse de sensibilité et optimisation pour le calage d'un modèle transport-urbanisme, http://papersjds16.sfds.asso.fr/submission_133.pdf.

Contributions à des logiciels

Nos contributions prennent la forme de deux fonctions implémentées dans le package `sensitivity` du logiciel R et d'un code Maple présenté dans l'annexe C.

- package `sensitivity` : Ce package contient plusieurs méthodes d'estimation de mesures de sensibilité, dont les indices de sensibilité de Sobol'. Nos contributions portent sur les fonctions `sobolEff` et `sobolroalhs` de ce package. Nous avons apporté une correction d'erreur à la fonction `SobolEff` dans l'implémentation des intervalles de confiance asymptotiques de l'estimateur (2.12) page 46. Nous avons développé et ajouté la fonction `sobolroalhs` au package. Cette fonction est une implémentation de la méthode répliquée, détaillée à la section 2.3.2 de ce manuscrit, permettant d'estimer les indices de Sobol' d'ordre un et fermés d'ordre deux dans le cadre de variables aléatoires indépendantes.
- code Maple : le code Maple développé permet de calculer différentes catégories d'indices de Sobol' : indices d'ordre un, indices fermés d'ordre deux et indices totaux, étant donné une fonction pour laquelle on dispose d'une expression analytique et pour des lois de distribution uniformes sur $[0, 1]$ des variables d'entrée supposées indépendantes.

Introduction

Contexte applicatif

Le développement et l'utilisation de modèles intégrés transport-urbanisme, plus connus sous le nom de modèles LUTI (Land Use Transport Integrated) dans la littérature anglo-saxonne, connaît un regain d'intérêt depuis une dizaine d'années. L'utilisation de modèles transport-urbanisme a pour objectif de modéliser les interactions complexes entre l'usage des sols et le transport de biens et de personnes sur un territoire. La taille du territoire étudié peut aller de l'échelle d'un village jusqu'à une région. Ces modèles sont principalement utilisés par des acteurs locaux de planification urbaine comme outils d'aide à la décision pour des politiques de développement urbain et d'aménagement des sols.

La majorité de ces modèles résulte d'un couplage entre différents modules simulant chacun divers phénomènes économiques. L'utilisation d'un modèle transport-urbanisme peut se décomposer en trois phases :

- i)* une phase dite d'instanciation durant laquelle les données nécessaires à la conception du modèle sont collectées auprès d'agences d'urbanisme ou directement dans des bases de données nationales,
- ii)* une phase dite de calage, où les paramètres propres au modèle sont ajustés afin que le modèle reproduise le plus fidèlement possible une ou plusieurs données,
- iii)* une phase dite de prospective, où différents scénarios de développement transport-urbain sont testés pour planifier l'aménagement des sols et/ou du réseau de transport.

Ce travail de thèse a été motivé et financé par le projet ANR CITiES dont l'un des objectifs principaux a été de développer des méthodologies de calage et des algorithmes efficaces pour l'estimation des paramètres des modèles de type transport-urbanisme. Nos résultats ont notamment permis de proposer une procédure de calage du modèle transport-urbanisme *Tranus* présenté au chapitre 6. Les collectivités locales et agences d'urbanisme sont souvent réticentes à utiliser les modèles transport-urbanisme, et ceci pour plusieurs raisons. D'une part, ces modèles mettent en jeu un nombre considérable de paramètres au travers d'équations souvent non linéaires. Certains de ces paramètres interviennent au sein d'équations classiques de théorie économique, de modèles de choix discrets, et peuvent ainsi être ajustés aisément via des techniques empruntées à l'économétrie. En revanche, une part non négligeable de ces paramètres correspond à des coefficients scalaires n'ayant pas de représentation économique particulière et qui sont pour la plupart très mal connus. D'autre part, à

notre connaissance, il n'existe à ce jour aucune procédure de validation pertinente.

Dans cette thèse, nous nous sommes restreints à l'étude du modèle transport-urbanisme Tranus pour lequel nous avons développé une procédure automatique de calage. La version calée de Tranus devrait par la suite être utilisée dans une phase de prospective, afin de tester des scénarios de développement transport-urbain conjointement avec l'agence d'urbanisme de Grenoble. La procédure que nous avons développée et implémentée pour le modèle Tranus combine une phase d'analyse de sensibilité, qui vise à détecter les paramètres les plus influents avec une phase d'optimisation permettant de caler ces derniers paramètres. La phase d'analyse de sensibilité a pour but de réduire la dimension du problème d'optimisation à résoudre ensuite. Déterminer les paramètres influents aura également un intérêt lors de la phase de prospective pour proposer des leviers pertinents pour remanier le paysage urbain notamment.

la suite de cette introduction est dédiée à une brève description des concepts d'analyse de sensibilité et de plans d'expériences. Nous énonçons ensuite les problématiques sous-jacentes à nos travaux de thèse et détaillons la composition de ce manuscrit. Nous commençons tout d'abord par définir le cadre mathématique de nos travaux.

Modèle numérique

L'étude d'un système complexe (par exemple un phénomène physique, mécanique, dans notre cas des phénomènes macro-économiques) passe par la conception d'un modèle numérique prenant la forme suivante :

$$y = f(\mathbf{x}),$$

où :

- f est le modèle numérique,
- $\mathbf{x}$ est un vecteur de variables d'état, encore appelées variables d'entrée, regroupant les différentes variables descriptives du modèle,
- y est une variable d'intérêt, encore appelée variable de sortie, correspondant à une grandeur d'intérêt du modèle.

Pour la suite de ce manuscrit nous abrégons par entrées et sorties les termes variables d'entrée et variables de sortie. Nous n'envisageons pas le cas d'entrées spatio-temporelles ni le cas d'entrées ou de sorties fonctionnelles. Nous n'envisageons pas non plus le cas de modèles numériques aléatoires pour lesquels un même jeu de valeurs des entrées peut donner lieu à différentes sorties du modèle. Ici, le modèle numérique considéré est un modèle dit déterministe, un même jeu de valeurs des entrées reproduit la sortie du modèle à l'identique.

Dans notre cas, le vecteur $\mathbf{x}$ regroupe les paramètres de Tranus dont la connaissance *a priori* est limitée. Nous nous intéressons à identifier les paramètres les plus

influent sur les sorties du modèle par leur effet respectif ou par un effet cumulé provenant d'une interaction avec un autre paramètre. Un outil mathématique répondant à cette problématique est l'analyse de sensibilité.

Analyse de sensibilité

L'analyse de sensibilité a pour objectif principal d'examiner comment une sortie d'un modèle numérique réagit à des variations des entrées du modèle. Dans le cadre de notre application sur Tranus, l'intérêt d'identifier les paramètres et interactions les plus influents est de pouvoir réduire le nombre de paramètres à ajuster dans la procédure de calage.

Les méthodes d'analyse de sensibilité peuvent être regroupées en trois grandes classes :

- i)* les méthodes dites de criblage qui classent les entrées par groupes suivant le degré d'influence sur la sortie du modèle. Ces méthodes sont qualitatives : elles ne permettent pas de comparer numériquement l'influence d'une entrée par rapport à une autre. Ces méthodes sont adaptées pour des modèles mettant en jeu un très grand nombre d'entrées (de plusieurs dizaines à une centaine).
- ii)* les méthodes locales qui étudient l'influence d'une entrée lorsqu'elle varie dans le voisinage d'une valeur nominale. Ces méthodes font souvent appel au calcul de dérivées partielles autour de cette valeur nominale.
- iii)* les méthodes globales qui, à l'opposé des méthodes locales, s'intéressent à l'ensemble du domaine de variation de l'entrée. L'influence d'une entrée est étudiée en même temps que les autres entrées varient. Ces méthodes reposent sur le calcul de résumés statistiques de la sortie du modèle quantifiant l'influence des entrées.

La méthode d'analyse de sensibilité considérée ici est une méthode globale reposant sur le calcul d'indicateurs de sensibilité appelés indices de Sobol'.

Indices de Sobol'

Les indices de Sobol' sont des grandeurs scalaires variant entre 0 et 1 déterminant quelle proportion de la variance de la sortie du modèle est due à telle entrée ou tel groupe d'entrées. L'influence d'une entrée (ou d'un groupe d'entrées) est quantifiée par la valeur de son indice respectif. Plus la valeur de cet indice est proche de 1, plus l'entrée (ou le groupe d'entrées) est influente.

Le calcul des indices de Sobol' permet ainsi d'ordonner les entrées suivant leur influence sur la sortie du modèle. Dans le cadre de notre application, il devient alors facile d'identifier les paramètres les plus influents et de détecter les potentielles interactions existantes. L'obtention de ces indices permet également un retour d'infor-

mation auprès du modélisateur, et autres utilisateurs avérés du modèle, confirmant ou infirmant des *a priori* portant sur l'influence relative des entrées.

Différentes approches existent pour calculer les indices de Sobol'. Dans notre cas, nous nous focalisons sur les approches de type Monte Carlo. Ces méthodes sont basées sur l'hypothèse d'indépendance des variables d'entrée. L'utilisation de telles méthodes se décompose en plusieurs étapes.

La première consiste à choisir le domaine de variation et la loi de distribution de chacune des entrées du modèle. Ces lois sont souvent connues du modélisateur. La deuxième étape consiste à échantillonner les entrées (ie sélectionner un ensemble de jeux de valeurs des entrées) suivant leur distribution et intervalle de variations. La troisième étape consiste à évaluer le modèle en chaque jeu de valeurs résultant de l'échantillonnage précédent. La dernière étape consiste à estimer les indices de Sobol' à partir de ces évaluations.

L'ensemble des jeux de valeurs, des points, obtenus à l'issue de la deuxième étape constitue ce que l'on appelle un plan d'expériences numériques.

Plan d'expériences

Introduit à l'origine pour la planification d'expériences physiques en laboratoire, le concept de plan d'expériences s'est élargi avec l'apparition des modèles numériques. La différence majeure réside dans le fait que les plans classiques construits en laboratoire supposent l'existence d'une erreur due aux mesures ou aux conditions expérimentales. Cette erreur n'existe pas dans le cas des plans d'expériences numériques.

Cependant, plans classiques et plans d'expériences numériques partagent un point commun. Dans les deux cas, l'objectif principal est d'extraire le plus d'information possible du modèle avec un nombre d'évaluations (de mesures) le plus petit possible.

La difficulté principale dans la construction d'un plan d'expériences numériques est d'échantillonner les variables d'entrée du modèle de façon optimale (en un sens à formaliser) avec peu, voir aucune, de connaissance *a priori* sur la sortie. Dans le cadre de l'estimation des indices de Sobol', la qualité du plan d'expériences numériques peut permettre de diminuer en moyenne le nombre d'évaluations nécessaires pour atteindre une précision donnée sur les indices estimés.

Problématiques

Parmi les approches Monte Carlo pour l'estimation des indices de Sobol', la méthode dite répliquée, présentée au chapitre 2, a l'avantage de permettre l'estimation de tous les indices d'ordre un pour un coût (ie le nombre total d'évaluations du modèle) indépendant de la dimension (ie le nombre de variables d'entrée) et de réduire considérablement le coût d'estimation des indices fermés d'ordre deux.

Guidé par notre application sur le modèle *Tranus*, les contributions de notre thèse portent principalement sur des extensions de la méthode répliquée pour répondre aux problématiques suivantes :

- P_1 : Comment gérer la présence de corrélations entre les variables d'entrée du modèle ?
- P_2 : Comment généraliser l'estimation des indices de Sobol' au cas de variables de sortie vectorielles ? Quels intervalles de confiance pour ces indices ?
- P_3 : Comment garantir un seuil de précision sur l'estimation des indices ? Intrinsèquement, comment réutiliser un premier jeu d'évaluations du modèle si la précision atteinte n'est pas satisfaisante ?

Apporter des réponses à ces questions a nécessité un travail important sur la construction de plans d'expériences stratifiés vérifiant des critères de remplissage de l'espace des entrées.

Plan de thèse

Le premier chapitre de ce manuscrit est consacré à une présentation détaillée des principaux types de plans d'expériences numériques ainsi qu'à des critères mesurant la qualité de ces plans. Le deuxième chapitre passe en revue plusieurs méthodes Monte Carlo d'estimation d'indices de Sobol', s'attardant plus particulièrement sur la méthode répliquée.

Les chapitres 3 à 6 regroupent nos contributions sous la forme de publications soumises ou acceptées apportant une réponse aux trois problématiques précédentes.

Le chapitre 3 propose d'étendre l'estimation des indices de Sobol' par la méthode répliquée au cas de groupes de variables d'entrée corrélées. La corrélation à l'intérieur de chaque groupe est définie par des contraintes dites linéaires ordonnées. Afin de satisfaire ces contraintes, la construction des deux plans répliqués fait intervenir une structure géométrique de type simplexe. Une stratégie dite *space-filling* est proposée pour construire ces plans.

Le chapitre 4 propose une version itérative de la méthode répliquée. L'utilisation d'une version itérative permet d'augmenter progressivement le nombre d'évaluations du modèle et d'affiner au fur et à mesure la précision des estimations des indices. Le point clé de cette approche, est la construction de plans répliqués emboîtés *space-filling*.

Le chapitre 5 propose une borne d'erreur pour l'estimation d'indices de Sobol' par échantillonnage quasi-Monte Carlo. La borne d'erreur est obtenue à partir d'une décomposition en coefficients de Walsh des intégrandes intervenant dans la formule intégrale de l'indice de Sobol'. Cette décomposition utilise la propriété de clôture d'un ensemble de points obtenu par méthodes d'échantillonnage quasi-Monte Carlo.

Nous proposons d'utiliser cette borne d'erreur comme critère d'arrêt pour deux méthodes d'estimation des indices.

Le chapitre 6 propose une généralisation de la méthode répliquée au cas de sortie vectorielle ainsi que la construction d'intervalles de confiance bootstrap des indices estimés. Cette méthode est appliquée au modèle *Tranus* comme première étape d'une procédure de calage des paramètres du modèle. S'ensuit une phase d'ajustement des paramètres identifiés comme les plus influents. La phase d'ajustement est une procédure itérative hybride au sens stochastique et déterministe.

Le manuscrit est également composé de trois annexes. Les annexes A et B apportent des premiers éléments de réponse à des questions très intéressantes soulevées par Art B. Owen lors de sa lecture d'une première version de ce manuscrit. L'annexe C présente un code pour le calcul des indices de Sobol' d'une fonction dont on dispose d'une expression analytique.

Notre manuscrit suivant le format de thèse dite par articles, les notations employées dans les prochains chapitres peuvent être amenées à varier d'un chapitre à un autre.

CHAPITRE 1

Plans d'expériences numériques : un tour d'horizon

Comme indiqué en introduction, le concept de plans d'expériences trouve ses racines dans les expériences dites physiques comme par exemple les expériences effectuées en laboratoire. Nous avons également succinctement évoqué en introduction la différence entre plans classiques et plans numériques. Nous proposons dans la partie 1.1 de repartir de la définition de plans d'expériences dits classiques tels qu'introduits par Fisher [Fis26] dans le cadre expérimental. Puis en s'appuyant sur cette définition, nous mettons en évidence les différences majeures entre plans classiques et plans numériques. La partie 1.2 passe en revue une classe spécifique de plans numériques appelés plans stratifiés (également appelés plans marginaux). Nous discutons ensuite des plans d'expériences issus de méthodes d'échantillonnage quasi-Monte Carlo. Nous nous limitons au cas des suites digitales que nous détaillons dans la partie 1.3. La dernière partie 1.4 est consacrée à la présentation de critères permettant d'évaluer la qualité des plans d'expériences numériques.

Pour la suite de ce chapitre, $\mathbf{x} = (x_1, \dots, x_d)$ désigne le vecteur des d variables d'entrée du modèle numérique f et $\mathcal{D} = \{1, \dots, d\}$. L'espace des entrées est restreint à l'hypercube $[0, 1]^d$. Les entrées $x_1, \dots, x_d$ sont modélisées par un vecteur de variables aléatoires indépendantes noté $\mathbf{X} = (X_1, \dots, X_d)$ et de loi uniforme sur $[0, 1]^d$. Cette dernière hypothèse n'est pas restrictive, la généralisation à d'autres lois produits peut être obtenue par diverses méthodes (*e.g.* transformée inverse, rejet) explicitées dans [Dev86].

L'incertitude sur les entrées est modélisée par $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,d})$, $i \in \mathbb{N}$. Un plan d'expérience consiste en un ensemble de points noté $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ que l'on représente également par une matrice de taille $n \times d$:

$$\mathcal{P} = \begin{pmatrix} X_{1,1} & \dots & X_{1,j} & \dots & X_{1,d} \\ \vdots & & \vdots & & \vdots \\ X_{i,1} & \dots & X_{i,j} & \dots & X_{i,d} \\ \vdots & & \vdots & & \vdots \\ X_{n,1} & \dots & X_{n,j} & \dots & X_{n,d} \end{pmatrix}.$$

n définit le nombre de lignes de la matrice, chaque ligne correspondant à un point $\mathbf{X}_i \in [0, 1]^d$. d définit le nombre de colonnes de la matrice, chaque colonne correspondant à une variable X_j , $j \in \mathcal{D}$. On modélise la variable de sortie y par une

variable aléatoire $Y = f(\mathbf{X})$. On suppose par la suite que Y est à valeurs dans $\mathbb{R}$ et que $f \in \mathbb{L}^2([0, 1]^d, \lambda(d\mathbf{x}))$ où λ désigne la mesure de Lebesgues sur $[0, 1]^d$. La valeur de la variable de sortie au point $\mathbf{X}_i$ est notée $Y_i = f(\mathbf{X}_i)$.

1.1 Des plans classiques aux plans numériques

Les prémices de l'utilisation de plans d'expériences remontent probablement aux expérimentations menées par Peirce [Pei78] afin d'étudier la capacité d'une personne à différencier ou non des poids les yeux bandés. La mise en place d'un formalisme et la conception d'une méthodologie pour la construction de plans d'expériences classiques sont introduites plus tard par Fisher [Fis26]. Bien que Fisher ait défini cette méthodologie dans le cadre de ses applications agricoles, elle sera reprise plus tard dans le cadre d'autres applications telles que des applications industrielles et militaires.

Dans le cadre des plans d'expériences classiques, les variables d'entrée sont appelées facteurs et la variable de sortie mesurée est appelée réponse. Une expérience correspond à un jeu de valeurs des facteurs et l'ensemble des expériences constitue le plan d'expériences. Les valeurs des facteurs sont appelées niveaux. Les objectifs de la construction d'un plan d'expériences classiques sont divers, par exemple :

- déterminer quels sont les facteurs qui par leur variation propre ou conjointe entraînent la plus grande variation mesurée de la réponse,
- déterminer si l'on peut linéariser la réponse en fonction des facteurs et si le modèle obtenu possède une bonne capacité prédictive.

Pour ce faire, dans un cas simple, la réponse est le plus souvent décomposée en un modèle linéaire prenant la forme :

$$\mathbf{Y} = \beta \mathbf{X}^T + \varepsilon, \quad (1.1)$$

où $\mathbf{Y}$ est le vecteur contenant les réponses des n expériences, $\beta = (\beta_0, \beta_1, \dots, \beta_d)$ est un vecteur de coefficients réels, $\mathbf{X}$ est une matrice de taille $n(d+1)$, T est l'opérateur transposé et $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$ est un vecteur modélisant l'erreur commise lors de chaque expérience. On suppose que la matrice de covariance de ε vaut $\Sigma_\varepsilon = \sigma^2 I_n$ où I_n est la matrice identité de taille $n \times n$.

Dans le modèle linéaire (1.1) le plus simple, $\mathbf{X}$ s'identifie au plan d'expériences $\mathcal{P}$ auquel on ajoute une première ligne remplie de 1. L'estimation des coefficients β_j , permet alors une interprétation des effets principaux des facteurs ; l'effet principal d'un facteur correspondant à l'effet de ce seul facteur sur la réponse du modèle. Pour des modèles plus complexes cherchant par exemple à prendre en compte des interactions, les termes de la matrice $\mathbf{X}$ sont des combinaisons des facteurs.

1.1.1 Principes fondamentaux

La caractéristique majeure des plans d'expériences classiques est la présence d'une erreur expérimentale : lors de l'expérimentation, l'obtention de la réponse

peut être sujette à des erreurs de mesure ou des erreurs dues aux conditions expérimentales. Tenant compte de cette caractéristique, Fisher formalise des principes fondamentaux pour construire des plans d'expériences plus robustes. Ces principes sont les suivant :

- i) la **répétition** consiste à répéter une expérience plusieurs fois afin d'estimer l'erreur expérimentale et de vérifier la fiabilité et la validité de l'expérience.
- ii) la **sélection aléatoire** (*random assignment*) est une méthode qui protège contre l'introduction d'un biais issu de l'ordre dans lequel les expériences sont menées.
- iii) le **blocking** est une méthode consistant à regrouper l'ensemble des expériences par paquet d'expériences similaires afin de réduire des sources de variations connues mais sans importance pour l'expérimentation.
- iv) l'**orthogonalité** fait référence à la propriété selon laquelle les vecteurs colonnes du plan d'expériences sont deux à deux orthogonaux (*ie* que leur produit scalaire est égal à zéro). Un plan d'expériences dit orthogonal permet d'estimer effets principaux et interactions entre facteurs sans effets de confusion (*confounding*).
- v) l'**expérimentation factorielle** consiste à privilégier des expériences factorielles (détaillées à la section 1.1.2) permettant d'évaluer à la fois les effets principaux et les potentielles interactions entre les facteurs.
- vi) la **flexibilité** traduit la capacité d'un plan d'expériences à pouvoir extraire de l'information pour différentes modélisations de la réponse.

Une grande variété de plans d'expériences classiques ont été conçus sur la base des principes de Fisher. Nous proposons de détailler brièvement une catégorie de plans classiques appelés plans factoriels. Pour une description exhaustive de l'éventail des plans classiques existants, nous renvoyons le lecteur aux synthèses de Box *et al.* [BHH78] et Box et Draper [BD87].

1.1.2 Plans factoriels complets et fractionnaires

Les plans d'expériences détaillés dans cette section sont obtenus en discrétisant indépendamment chaque facteur X_j , $j \in \mathcal{D}$, en un petit nombre de niveaux.

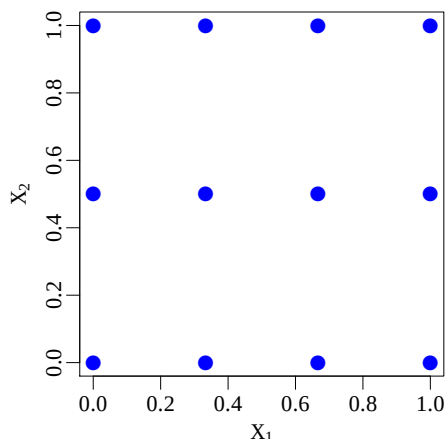
Un plan factoriel complet est construit de façon à inclure toutes les combinaisons possibles des niveaux des facteurs, chaque combinaison contenant de l'information sur plusieurs facteurs à la fois. Ce type de plans est flexible et efficace pour étudier en même temps effets principaux et interactions entre facteurs.

La taille du plan factoriel complet correspond au produit du nombre de niveaux de chaque facteur. Par exemple, un plan factoriel complet à trois facteurs ayant respectivement 2, 3 et 4 niveaux contient un total de 24 combinaisons. Si les facteurs ont tous un même nombre de niveaux, la taille du plan s'écrit alors l^d où l désigne le nombre de niveaux et d le nombre de facteurs.

La figure 1.1 représente les points d'un plan factoriel complet à deux facteurs avec

niveaux respectifs $\{0, 1/3, 2/3, 1\}$ et $\{0, 1/2, 1\}$. La taille d'un plan factoriel complet

FIGURE 1.1 – Plan factoriel complet à deux facteurs et 12 combinaisons.



augmente exponentiellement en fonction du nombre de facteurs. Pour un nombre élevé de facteurs, il devient impossible de tester toutes les combinaisons existantes. Une alternative consiste à avoir recours à un plan factoriel dit fractionnaire.

Un plan factoriel fractionnaire est composé d'un sous ensemble des combinaisons d'un plan factoriel complet. Ces plans permettent d'estimer les effets principaux et les interactions d'ordre faible (interactions entre un petit nombre de facteurs) avec un nombre de combinaisons considérablement réduit en sacrifiant la possibilité d'estimer des interactions d'ordre élevé (interactions entre un nombre élevé de facteurs). Pour plus de détails sur ces plans, le lecteur pourra consulter les travaux de Myers et Montgomery [MM95].

Les plans factoriels fractionnaires peuvent être catégorisés suivant un indicateur appelé niveau de résolution. Le niveau de résolution renseigne sur l'ordre maximum d'interactions que peut estimer un plan d'expériences sans entraîner de confusion (*confounding*) entre effets dus aux interactions et effets principaux des facteurs.

1.1.3 Vers les plans d'expériences numériques

Les plans d'expériences numériques cherchent à répondre au même objectif que les plans d'expériences classiques : obtenir le plus d'informations possible sur le modèle. Cependant des différences majeures existent entre ces deux types de plans. Les plans d'expériences numériques sont construits pour simuler une réponse d'un modèle numérique déterministe. Au contraire du modèle (1.1), les modèles déterministes sont caractérisés par l'absence d'un terme d'erreur ε . Le modèle évalué en un point donne toujours la même réponse. L'absence du terme d'erreur est à l'origine des différences suivantes :

- certains principes énoncés par Fisher pour la construction de plans d'expériences classiques n'ont plus lieu d'être dans le cadre de la construction d'un

plan d'expériences numériques : répétition, sélection aléatoire, blocking.

- Un modèle linéaire tel que (1.1) est très souvent inadapté pour la modélisation de systèmes numériques complexes.
- dans le cas des modèles numériques, le nombre de variables d'entrée est en principe plus élevé.
- le domaine de variation des entrées est plus large car, sauf exception, aucune contrainte expérimentale ne permet de réduire l'ensemble des valeurs des entrées à des sous-domaines.

En général, comme les modèles numériques mettent en jeu des systèmes d'équations complexes et que le comportement de la sortie n'est pas connu à l'avance, les plans d'expériences numériques dont la construction requiert peu d'hypothèses sur le modèle paraissent être les mieux adaptés. Les principes de flexibilité et d'orthogonalité énoncés par Fisher restent valables pour la conception de plans d'expériences numériques. De même l'utilisation de plans factoriels est envisageable.

1.2 Plans stratifiés

La construction d'un plan d'expériences numériques fait appel à une méthode d'échantillonnage (de tirage) pour sélectionner les points en lesquels le modèle sera évalué. La méthode d'échantillonnage la plus simple consiste à générer indépendamment pour chaque variable X_j un ensemble de n réalisations indépendantes d'une variable aléatoire de loi uniforme sur $[0, 1]$. Plus formellement, l'ensemble des points résultant d'un tel échantillonnage est noté $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ avec :

$$\mathbf{X}_i = (U_{i,1}, \dots, U_{i,d}) , \quad i \in \{1, \dots, n\},$$

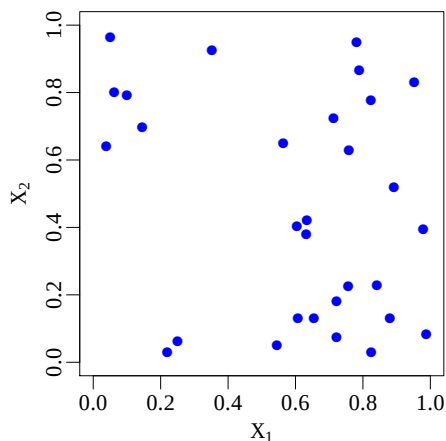
où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$. Un tel échantillonnage est qualifié d'échantillonnage aléatoire, le plan d'expériences $\mathcal{P}$ en résultant est appelé plan aléatoire.

Lorsqu'on travaille à n fixé, un tel plan ne donne aucune garantie quant à une bonne répartition des points dans l'espace. Un plan aléatoire fait très souvent apparaître à la fois des zones lacunaires et des zones où se concentrent les points échantillonnés. Ce phénomène est illustré sur la figure 1.2 pour une dimension $d = 2$ et un nombre de points $n = 30$.

Ce type de problèmes peut être amélioré en considérant des plans d'expériences issus de méthodes d'échantillonnage consistant à partitionner l'espace des entrées en sous-domaines distincts (d'intersection vide deux à deux). Une fois ce découpage effectué, un point est échantillonné aléatoirement dans chaque sous-domaine.

Les plans que nous décrivons dans cette section reposent sur cette idée de découpage et sont appelés plans stratifiés. Le terme strate réfère à un des sous-domaines issu du découpage. Nous nous focalisons en particulier sur les cas de grilles régulières, hypercubes latins et tableaux orthogonaux.

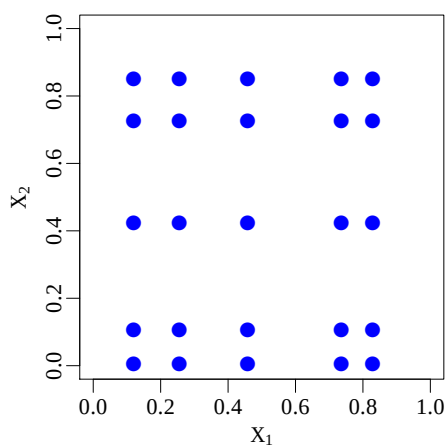
FIGURE 1.2 – Plan aléatoire à 30 points en dimension 2.



1.2.1 Grilles régulières

Une grille régulière est obtenue en discrétisant indépendamment chaque variable $X_j, j \in \mathcal{D}$, en un ensemble de n valeurs distinctes puis en formant les n^d combinaisons possibles. La figure 1.3 illustre un exemple d'une grille régulière à 25 points en dimension 2. Une grille régulière correspond à un plan factoriel complet à d facteurs

FIGURE 1.3 – Grille régulière à 25 points en dimension 2.



où chaque facteur possède n niveaux distincts.

Les points d'une grille régulière sont alignés le long d'axes identifiés par les n valeurs choisies pour chaque variable. Il est possible de casser cette structure en considérant le plan suivant :

Définition 1.1. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ l'ensemble de points issus de la grille régulière construite en considérant pour chaque variable $X_j, j \in \mathcal{D}$, le même jeu de n valeurs

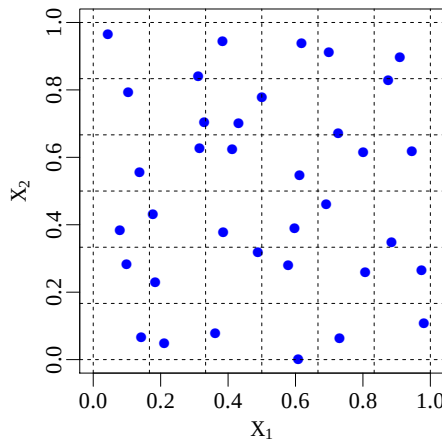
$\{1/n, 2/n, \dots, 1\}$. Soit l'ensemble de points $\mathcal{P}_0 = \{\mathbf{Z}_i\}_{i=1}^n$ construit par :

$$\mathbf{Z}_i = (X_{i,1} - U_{i,1}/n, \dots, X_{i,d} - U_{i,d}/n), \quad i \in \{1, \dots, n\}, \quad (1.2)$$

où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$. Alors $\mathcal{P}_0$ est une grille régulière randomisée.

La figure 1.4 illustre une grille régulière randomisée $\mathcal{P}_0$ à 6^2 points construite suivant (1.2). Une grille régulière randomisée est obtenue en faisant varier chaque

FIGURE 1.4 – Grille régulière randomisée à 6^2 points.



point d'une grille régulière indépendamment les uns des autres à l'intérieur de leur strate respective. Ces types de plans stratifiés sont adaptés uniquement dans le cas de petites dimensions, 2 ou 3. Pour des plus grandes valeurs de d , le nombre de points créés (n^d) croît exponentiellement et devient rapidement inadapté aux modèles requérant un temps d'évaluation important.

Une alternative consiste à sélectionner un sous-ensemble des points $\mathbf{Z}_i$ créés suivant l'équation (1.2), de façon à assurer une bonne répartition des valeurs prises par chaque variable X_j . Une catégorie de plans d'expériences résultant d'une telle approche correspond aux hypercubes latins.

1.2.2 Hypercubes latins

Les hypercubes latins ont été introduits dans la littérature par McKay [MBC79] dans le contexte de l'évaluation numérique d'intégrales multiples. Un hypercube latin est défini comme suit :

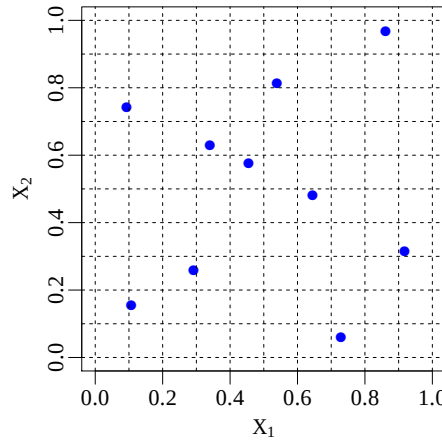
Définition 1.2. Notons par Π_n l'ensemble des permutations de $\{1, \dots, n\}$. Soient $\pi_1, \dots, \pi_d$, d variables aléatoires indépendantes uniformément distribuées sur Π_n . Un hypercube latin $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ est défini par :

$$\mathbf{X}_i = \left(\frac{\pi_1(i) - U_{i,1}}{n}, \dots, \frac{\pi_d(i) - U_{i,d}}{n} \right), \quad i \in \{1, \dots, n\}, \quad (1.3)$$

où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$. $(\pi_1(i)/n, \dots, \pi_d(i)/n)$ renseigne sur la strate à laquelle le point $\mathbf{x}_i$ appartient, $(U_{i,1}, \dots, U_{i,d})$ indique à quel endroit le point est situé dans la strate.

La figure 1.5 illustre un hypercube latin à 10 points en dimension 2. Le carré $[0, 1]^2$ peut être segmenté horizontalement ou verticalement en 10 segments (lignes pointillées) et chaque segment contient un unique point. Ainsi, un hypercube latin garantit une bonne répartition des points dans les projections d'ordre un.

FIGURE 1.5 – Hypercube latin à 10 points en dimension 2.



Un hypercube latin peut également être défini dans sa version centrée pour laquelle les points sont alors placés au centre des strates. L'équation (1.3) devient alors :

$$\mathbf{X}_i = \left(\frac{\pi_1(i) - 1/2}{n}, \dots, \frac{\pi_d(i) - 1/2}{n} \right), \quad i \in \{1, \dots, n\}.$$

Nous énonçons à présent quelques propriétés relatives aux hypercubes latins en commençant par un théorème portant sur la distribution des points $\mathbf{x}_i$:

Théorème 1.1. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ un hypercube latin défini par (1.3), alors pour chaque $i \in \{1, \dots, n\}$, $\mathbf{X}_i$ est uniformément distribuée sur $[0, 1]^d$.

Démonstration. Nous renvoyons à la preuve du théorème 10.1 dans [Owe13b]. $\square$

Soit un modèle numérique $f \in \mathbb{L}^2([0, 1]^d)$ dont nous notons par $\mu = \int_{[0, 1]^d} f(\mathbf{x}) d\mathbf{x}$ sa moyenne et $\sigma^2 = \int_{[0, 1]^d} (f(\mathbf{x}) - \mu)^2 d\mathbf{x}$ sa variance. L'estimateur de μ par un hypercube latin s'écrit :

$$\hat{\mu}_{lhs} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i).$$

Une conséquence immédiate du théorème 1.1 est que $E[\hat{\mu}_{lhs}] = \mu$. Pour une fonction $f \in \mathbb{L}^2([0, 1]^d)$, Stein [Ste87] établit l'expression suivante de la variance de $\hat{\mu}_{lhs}$:

Proposition 1.1. Soit $f \in \mathbb{L}^2([0, 1]^d)$ et $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$, $n \geq 2$, un hypercube latin défini par (1.3). Alors :

$$\text{Var}[\hat{\mu}_{l_{hs}}] = \frac{1}{n} \int_{[0,1]^d} e(\mathbf{x})^2 d\mathbf{x} + o\left(\frac{1}{n}\right),$$

où $e(\mathbf{x}) = f(\mathbf{x}) - f^{add}(\mathbf{x})$, f^{add} désignant la partie additive de f .

Ainsi, pour des fonctions presque additives, l'utilisation d'un hypercube latin offre une réduction considérable de la variance.

Toujours dans [Ste87], Stein compare asymptotiquement les variances $\text{Var}[\hat{\mu}_{l_{hs}}]$ et $\text{Var}[\hat{\mu}_{iid}]$, où $\hat{\mu}_{iid}$ désigne l'estimateur de μ par un plan aléatoire. Le résultat obtenu est que $\text{Var}[\hat{\mu}_{l_{hs}}]$ est asymptotiquement toujours plus petite que $\text{Var}[\hat{\mu}_{iid}]$. Ce résultat est généralisé dans [Loh96] au cas de sorties multidimensionnelles. Toujours dans [Loh96], Loh prouve l'existence d'un théorème central limite pour un hypercube latin sous la condition $f \in \mathbb{L}^3([0, 1]^d)$.

Dans [Owe97a], Owen donne un résultat non asymptotique pour $\text{Var}[\hat{\mu}_{l_{hs}}]$ à taille n fixe :

Proposition 1.2. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$, $n \geq 2$, un hypercube latin défini par (1.3), et $f \in \mathbb{L}^2([0, 1]^d)$ alors $\text{Var}[\hat{\mu}_{l_{hs}}] \leq \sigma^2/(n-1)$.

Démonstration. Nous renvoyons à la preuve de la proposition 3 dans [Owe97a]. $\square$

Sachant que $\text{Var}[\hat{\mu}_{iid}] = \sigma^2/n$, la proposition 1.2 nous dit que dans le pire des cas $\hat{\mu}_{l_{hs}}$ a une variance seulement $n/(n-1)$ fois plus grande que celle de $\hat{\mu}_{iid}$.

Dans une perspective d'analyse de sensibilité, la stratification raffinée de chaque variable X_j , $j \in \mathcal{D}$, induite par un hypercube latin est une propriété souhaitable. Une variable X_j influente sur la sortie a alors plus de chances d'être identifiée. Toutefois, un hypercube latin ne garantit pas nécessairement une bonne répartition des points dans le domaine $[0, 1]^d$ ou dans des projections d'ordre supérieur à un. Par exemple, une mauvaise configuration correspond à obtenir tous les points alignés dans la diagonale de $[0, 1]^d$.

Des alternatives ont été proposées pour construire de « meilleurs » hypercubes latins. Une de ces alternatives consiste à construire des hypercubes latins pour lesquels les corrélations entre les variables X_j sont petites. On citera entre autre les travaux de Iman et Conover [IC82] et Kenny [Ye98]. Une deuxième alternative consiste à construire des hypercubes latins satisfaisant des critères de bon remplissage de l'espace. Nous présentons certains de ces critères dans la section 1.4.

1.2.3 Tableaux orthogonaux

Le concept de tableau orthogonal a été introduit par Kishen [Kis42] puis repris plus tard par Rao [Rao46]. Un tableau orthogonal peut être vu comme la généralisation d'un hypercube latin à des projections d'ordre supérieur. Tandis qu'un

hypercube latin garantit une bonne stratification de chaque variable X_j , $j \in \mathcal{D}$, un tableau orthogonal garantit lui une bonne stratification de chaque sous-ensemble de variables $\{X_j\}_{j \in u}$, $u \subset \mathcal{D}$, pour $t = \#u$ fixé où $\#$ est l'opérateur cardinalité et $\subset$ désigne l'inclusion stricte. Dans [HSS99], Hedayat définit un tableau orthogonal de la manière suivante :

Définition 1.3. Soit $(d, q, t) \in \mathbb{N}^3$ tel que $q \geq 2$ et $d \geq t \geq 1$. Soit S un ensemble de cardinalité q . Un tableau orthogonal est une matrice de taille $n \times d$ dont les éléments appartiennent à S , telle que chaque sous matrice de taille $n \times t$ contient les q^t sous-ensembles de S à t éléments λ fois.

Un tableau orthogonal est noté $OA(n, d, q, t)$ où t désigne la force du tableau orthogonal, q son nombre de niveaux (éléments de S) et λ son index. Le nombre de lignes du tableau est $n = \lambda q^t$. Ici, l'ensemble S est identifié par le corps de Galois à q éléments noté $GF(q)$. Pour une description complète des méthodes et résultats concernant les tableaux orthogonaux, nous renvoyons le lecteur au livre d'Hedayat [HSS99]. Nous nous focalisons ici sur une classe particulière de tableaux orthogonaux appelés tableaux orthogonaux linéaires :

Définition 1.4. Un tableau orthogonal $OA(n, d, q, t)$ est dit linéaire si $\lambda = 1$ et si q est une puissance d'un nombre premier c'est à dire $q = p^k$ où p est un nombre premier et $k \in \mathbb{N}$, $k \geq 1$. Les lignes d'un tableau orthogonal linéaire forment un sous-espace vectoriel de $GF(q)$.

La terminologie linéaire provient de la propriété que chaque sous-ensemble de t colonnes d'un $OA(n, d, q, t)$ linéaire est linéairement indépendant dans $GF(q)$. Quand $t = 1$, le tableau orthogonal linéaire devient un hypercube latin. Quand $t = d$ le tableau orthogonal linéaire devient une grille régulière. Plus généralement, la structure d'un tableau orthogonal linéaire de force t garantit une bonne répartition des points dans les projections d'ordre t .

L'existence de tableaux orthogonaux n'est pas assurée pour un jeu de paramètres (d, q, t) quelconque. Dans [Rao47], plusieurs résultats de limite d'existence des tableaux orthogonaux sont donnés. Dans [Bus52], Bush étend ces résultats et démontre le théorème d'existence suivant :

Théorème 1.2. Soit $q = p^k$, avec $p \geq 2$ premier et $k \geq 1$ entier naturel, alors pour $t \geq 1$ vérifiant $t \leq q - 1$ le tableau orthogonal $OA(q^t, q + 1, q, t)$ existe.

Démonstration. Nous renvoyons à la preuve du théorème 3.1 dans [HSS99]. □

En parallèle du théorème 1.2, Bush propose une construction d'un tableau orthogonal $OA(q^t, q + 1, q, t)$. Cette construction consiste à évaluer q^t polynômes distincts avec les q éléments de $GF(q)$, les opérations étant menées dans $GF(q)$. Dans le cas particulier où $t = 2$, une construction alternative repose sur l'utilisation de matrices aux différences de force 2, faciles à implémenter. Cette dernière construction est détaillée dans [HSS99, Lemma 6.12].

Pour $k = 1$, ces deux constructions sont simples à implémenter grâce à l'isomorphisme entre $GF(q)$ et $\mathbb{Z}/q\mathbb{Z}$ (entiers naturels modulo q). Pour $q = p^k$ et $k > 1$, ces constructions deviennent plus complexes et nécessitent de faire appel à des logiciels de calcul formel.

Considérons un tableau orthogonal $OA(q^t, q + 1, q, t)$ noté $A = \{\mathbf{A}_i\}_{i=1}^{q^t}$ avec $\mathbf{A}_i = (A_{i,1}, \dots, A_{i,d})$, $A_{i,j} \in GF(q)$. Afin de construire un plan d'expériences $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^{q^t}$ à partir de A , les q niveaux de A sont substitués par $\{1, \dots, q\}$ (à l'aide d'une bijection de $GF(q)$ dans $\{1, \dots, q\}$). $\mathcal{P}$ est ensuite obtenu par :

$$\mathbf{X}_i = \left(\frac{A_{i,1} - U_{A_{i,1},1}}{q}, \dots, \frac{A_{i,d} - U_{A_{i,1},d}}{q} \right), \quad i \in \{1, \dots, q^t\}, \quad (1.4)$$

où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$.

Cette approche a un défaut majeur. Le plan d'expérience $\mathcal{P}$ construit par l'équation (1.4) hérite de la structure de sous-espace vectoriel de A . Par conséquent, les points $\mathbf{X}_i$ sont situés le long de plans dans $[0, 1]^d$ faisant apparaître de larges zones lacunaires. Une solution à ce problème consiste à détruire la structure de sous-espace vectoriel en considérant un tableau orthogonal dit randomisé :

Définition 1.5. Soit $A = \{\mathbf{A}_i\}_{i=1}^{q^t}$ un tableau orthogonal $OA(q^t, q + 1, q, t)$. Soient $\pi_1, \dots, \pi_d$, d variables aléatoires indépendantes uniformément distribuées sur Π_q . Un ensemble de points $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^{q^t}$ construit à partir d'un tableau orthogonal randomisé est défini par :

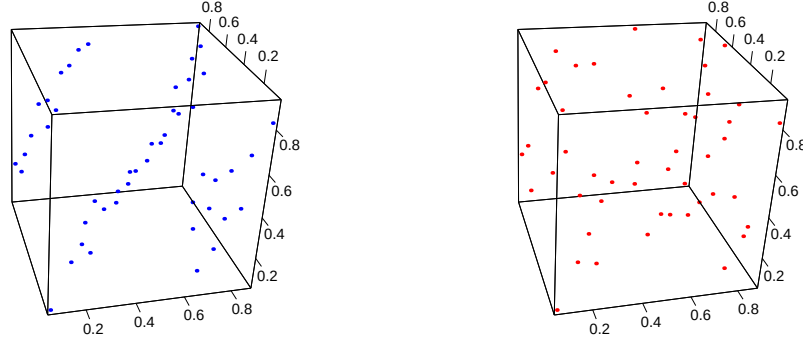
$$\mathbf{X}_i = \left(\frac{\pi_1(A_{i,1}) - U_{\pi_1(A_{i,1}),1}}{q}, \dots, \frac{\pi_d(A_{i,d}) - U_{\pi_d(A_{i,d}),d}}{q} \right), \quad i \in \{1, \dots, q^t\}, \quad (1.5)$$

où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$.

Par abus de langage, l'ensemble de points $\mathcal{P}$ construit dans la définition 1.5 est appelé tableau orthogonal randomisé. La figure 1.6 illustre la différence entre les constructions (1.4) et (1.5) pour un tableau orthogonal $OA(7^2, 3, 7, 2)$ et un même ensemble de variables aléatoires $U_{i,j}$. Notons que la définition que l'on donne d'un tableau orthogonal randomisé diffère de celle donnée par Owen dans [Owe92] où le terme aléatoire retranché dans (1.5) est $U_{i,j}$ au lieu de $U_{\pi_j(A_{i,j}),j}$. Ici, nous cherchons à conserver l'alignement des points dans les projections d'ordre t .

Un tableau orthogonal randomisé satisfait le théorème 1.1 énoncé pour un hypercube latin. Dans [Owe94], Owen donne une expression de la variance de μ estimée à l'aide d'un tableau orthogonal randomisé. Le résultat met en évidence une variance plus petite que celle de $\hat{\mu}_{lhs}$. Plus récemment dans [Loh08], Low démontre un théorème central limite pour les tableaux orthogonaux randomisés de force 2 sous des hypothèses de régularité de f .

FIGURE 1.6 – Un ensemble de points $\mathcal{P}$ construit par (1.4) pour un tableau orthogonal $OA(7^2, 3, 7, 2)$ et sa version randomisée construite par (1.5).



Dans une perspective d'analyse de sensibilité, la bonne stratification des projections d'ordre t induite par un tableau orthogonal linéaire de force t permet d'estimer avec plus de précision les interactions d'ordre correspondant. Pour un tableau orthogonal linéaire de force t , les variables X_j , $j \in \mathcal{D}$, sont chacune seulement stratifiées en q intervalles. Ainsi, pour obtenir une bonne stratification de chaque variable X_j , il est nécessaire de prendre une valeur de q suffisamment grande. Un inconvénient est que cela peut aboutir à une grande taille (q^t) du plan d'expériences pour des valeurs de t élevées.

Des alternatives ont été proposées pour construire de « meilleurs » tableaux orthogonaux. On citera entre autre les récents travaux de He et Tang [HT13, HT14] portant sur la construction de tableaux orthogonaux de force t dits *strong* assurant une meilleure répartition des points dans les projections d'ordre g , $g \leq t - 1$.

Dans le cadre de nos travaux, nous nous limitons à l'utilisation de tableaux orthogonaux randomisés de force $t = 2$ pour lesquels la taille n'est pas contraignante. Nous terminons cette section par la présentation d'une structure alliant tableau orthogonal et hypercube latin.

1.2.4 Hypercubes latins basés sur tableaux orthogonaux

Tang propose dans [Tan93] de combiner la structure de tableau orthogonal avec celle d'hypercube latin dans ce qu'il nomme hypercube latin basé sur tableau orthogonal. La définition d'une telle structure est la suivante :

Définition 1.6. Soit A un tableau orthogonal $OA(n, d, q, t)$. Pour chaque colonne de A , les λq^t éléments de niveau $k \in \{1, \dots, q\}$ sont remplacés par une permutation de l'ensemble $\{(k-1)\lambda q^{t-1} + 1, (k-1)\lambda q^{t-1} + 2, \dots, k\lambda q^{t-1}\}$. Soit $B = \{\mathbf{B}_i\}_{i=1}^n$ le tableau résultant. Un hypercube latin basé sur tableau orthogonal est l'ensemble de

points $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ défini par :

$$\mathbf{X}_i = \left(\frac{B_{i,1} - U_{i,1}}{n}, \dots, \frac{B_{i,d} - U_{i,d}}{n} \right), \quad i \in \{1, \dots, n\}, \quad (1.6)$$

où les $U_{i,j}$, $j \in \mathcal{D}$, sont des variables aléatoires indépendantes uniformément distribuées sur $[0, 1]$.

Si le tableau orthogonal utilisé n'est pas randomisé, l'hypercube latin obtenu hérite de la structure de sous-espace vectoriel du tableau orthogonal et de ses inconvénients. Ainsi en pratique il est souvent de mise de randomiser au préalable le tableau orthogonal A . L'hypercube latin basé sur tableau orthogonal jouit d'une bonne stratification des projections d'ordre 1 et t . Toutefois, dans le cadre de nos travaux portant sur l'estimation d'indices de Sobol', nous n'avons pas observé d'améliorations significatives par rapport à l'utilisation d'un hypercube latin.

Dans [Tan93], Tang généralise l'expression de $\text{Var}[\hat{\mu}_{l_{hs}}]$, énoncée dans la proposition 1.1, au cas des hypercubes latins basés sur tableaux orthogonaux linéaires de force 2.

1.3 Suites digitales et réseaux digitaux

Cette section s'inspire principalement de la synthèse sur les suites digitales proposée par Lemieux dans [Lem09]. On considère ici un ensemble de points $\mathcal{P} = \{\mathbf{X}_i\}_{i=0}^{n-1}$ dont on fait varier l'index i de 0 à $n - 1$ au lieu de 1 à n comme dans les sections précédentes. On débute cette section en se plaçant en dimension un, c'est à dire $\mathbf{X}_i \in [0, 1]$.

L'idée des suites digitales et réseaux digitaux est de construire les points $\mathbf{X}_i$ à partir d'une décomposition de l'index i dans une base $b \in \mathbb{N}$, $b \geq 2$. Plus formellement, la décomposition en base $b \geq 2$ d'un entier $i \geq 0$ s'écrit :

$$i = \sum_{k=0}^{\infty} i_k b^k, \quad i_k \in \{0, \dots, b-1\}, \quad (1.7)$$

où l'on suppose qu'un nombre fini de coefficients i_k seulement sont non nuls. Pour obtenir $\mathbf{X}_i$ à partir de la décomposition (1.7) de i , on utilise la fonction radicale inverse en base b , notée ϕ_b et définie par :

$$\phi_b(i) = \sum_{k=0}^{\infty} i_k b^{-k-1}, \quad i_k \in \{0, \dots, b-1\}. \quad (1.8)$$

On a naturellement $\phi_b(i) \in [0, 1]$. La suite digitale la plus simple et pionnière du concept est la suite de Van der Corput introduite dans [dC35]. Le point $\mathbf{X}_i$ de la suite de Van der Corput est donné par $\phi_b(i)$. Par exemple, en base $b = 4$, les 6 premiers points de la suite de Van der Coput sont :

$$(\mathbf{X}_0 = 0, \mathbf{X}_1 = 1/4, \mathbf{X}_2 = 1/2, \mathbf{X}_3 = 3/4, \mathbf{X}_4 = 1/16, \mathbf{X}_5 = 5/16).$$

Les 4 premiers points sont espacés d'une distance de $1/4$. Puis les 2 nouveaux points $\mathbf{X}_4, \mathbf{X}_5$ sont placés chacun dans les 2 intervalles formés par les 3 premiers points et sont à nouveau espacés d'une distance $1/4$.

La suite de Van der Coput, et plus généralement les suites digitales en dimension $d > 1$, possèdent deux avantages principaux sur les plans numériques présentés dans la section 1.2. Le premier est que la construction est naturellement itérative. Etant donné un premier jeu de b^m points, les b^m prochains points sont obtenus à partir des entiers $i = b^m, \dots, b^{m+1} - 1$. Le second concerne le remplissage de l'espace, les points $\mathbf{x}_i$ sont construits dans l'optique de combler les zones lacunaires de $[0, 1]^d$.

Les concepts de réseaux digitaux et suites digitales ont été introduits par Niederreiter [Nie92a], dans le contexte de l'intégration numérique afin de définir des points possédant de bonnes propriétés d'uniformité dans $[0, 1]^d$. Réseaux digitaux et suites digitales apparaissent dans la littérature sous les noms de (t, m, d) -réseaux digitaux et (t, d) -suites digitales abrégés par (t, m, d) -réseaux et (t, d) -suites. Nous détaillons ces deux concepts dans la section suivante.

1.3.1 (t, m, d) -réseaux et (t, d) -suites

Nous débutons cette section par la définition d'intervalle (ou pavé) élémentaire :

Définition 1.7. *Un intervalle élémentaire est un sous-ensemble de $[0, 1]^d$ défini par :*

$$J(\boldsymbol{\alpha}) = \prod_{j=1}^d \left[\frac{\alpha_j}{b^{\gamma_j}}, \frac{\alpha_j + 1}{b^{\gamma_j}} \right], \quad (1.9)$$

où $(\alpha_j, \gamma_j) \in \mathbb{N}^2$ tel que $\alpha_j < b^{\gamma_j}$ et $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d)$.

Un intervalle élémentaire a pour volume $b^{-\gamma}$ où $\gamma = \sum_{j=1}^d \gamma_j$. Soit $b \geq 2$, on note par $\mathcal{A}$ l'ensemble de tous les intervalles élémentaires $J(\boldsymbol{\alpha})$ de $[0, 1]^d$ définis par (1.9). Nous pouvons maintenant définir la structure de (t, m, d) -réseau :

Définition 1.8. *Pour $b \geq 2$ et $(m, t) \in \mathbb{N}^2$ tels que $m \geq t > 0$, l'ensemble des points $\mathcal{P} = \{\mathbf{X}_i\}_{i=0}^{b^m-1}$ est un (t, m, d) -réseau en base b si chaque intervalle élémentaire $J(\boldsymbol{\alpha}) \in \mathcal{A}$ de volume b^{t-m} contient b^t points de $\mathcal{P}$.*

Ainsi, un (t, m, d) -réseau en base b est défini de sorte que chaque intervalle élémentaire $J(\boldsymbol{\alpha})$ de volume b^{t-m} contienne un même nombre de points de $\mathcal{P}$. Les (t, m, d) -réseaux assurant la meilleure répartition des points dans $[0, 1]^d$ sont ceux pour lesquels $t = 0$. En effet, pour un $(0, m, d)$ -réseau, $[0, 1]^d$ peut être partitionné par les intervalles élémentaires de plus petit volume, b^{-m} , chacun contenant un point. La qualité d'un (t, m, d) -réseau est donc mesurée par son paramètre t appelé t -valeur.

En augmentant m , on augmente le nombre de points du (t, m, d) -réseau. La définition de (t, d) -suite s'obtient en faisant tendre m vers $+\infty$:

Définition 1.9. Pour $b \geq 2$ et $(m, t) \in \mathbb{N}^2$, la suite de points $\{\mathbf{X}_i\}_{i \in \mathbb{N}}$ est une (t, d) -suite en base b si, $\forall k \geq 0$ et $\forall m \geq t > 0$, chaque sous-ensemble de points $\{\mathbf{X}_i\}_{i=k b^m}^{(k+1)b^{m+1}-1}$ est un (t, m, d) -réseau en base b .

Nous proposons d'illustrer le concept de (t, m, d) -réseau. Considérons un $(0, 3, 2)$ -réseau de 2^3 points en base 2. D'après la définition 1.8 le carré $[0, 1]^2$ peut être partitionné par les intervalles élémentaires de volume 2^{-m} avec $m = 3$. D'après la définition 1.7, ces intervalles sont de la forme :

$$\left[\frac{\alpha_1}{2^{\gamma_1}}, \frac{\alpha_1 + 1}{2^{\gamma_1}} \right] \times \left[\frac{\alpha_2}{2^{\gamma_2}}, \frac{\alpha_2 + 1}{2^{\gamma_2}} \right],$$

avec $\gamma_1 + \gamma_2 = 3$, $\alpha_1 < 2^{\gamma_1}$ et $\alpha_2 < 2^{\gamma_2}$. La forme de ces intervalles dépend du choix que l'on fait du couple (γ_1, γ_2) . Il y a 4 configurations possibles de ce couple : $(0, 3)$, $(1, 2)$, $(2, 1)$ et $(3, 0)$. Pour chacun de ces couples, en faisant varier α_1 et α_2 respectivement dans $\{0, \dots, 2^{\gamma_1}\}$ et $\{0, \dots, 2^{\gamma_2}\}$ on obtient une partition de $[0, 1]^2$. Les 4 partitions sont représentées dans la figure 1.7.

Ainsi, les (t, m, d) -réseaux jouissent d'une stratification multiple garantissant une bonne répartition des points dans les projections d'ordre 1. La qualité de la répartition des points du réseau dans $[0, 1]^d$ est conditionnée par la t -valeur.

A l'origine, le critère t -valeur a été introduit par Sobol' afin de caractériser l'uniformité de ses suites digitales appelées suites de Sobol' [Sob67]. Nous présentons dans la section qui suit ces suites.

1.3.2 Suites de Sobol'

Les suites digitales en base b sont engendrées par un ensemble de matrices, appelées matrices génératrices, construites sur le corps de Galois $GF(b) = \{0, \dots, b-1\}$. Les suites de Sobol' sont des suites digitales en base $b = 2$. Pour une dimension $d \geq 1$, on note $C_1, \dots, C_d$ les d matrices génératrices d'une suite de Sobol'. Ces matrices sont des matrices triangulaires supérieures de rang plein construites sur $GF(2)$. La construction des matrices $C_1, \dots, C_d$ est itérative. Nous détaillons ci-dessous la construction de C_j pour $j \in \mathcal{D}$.

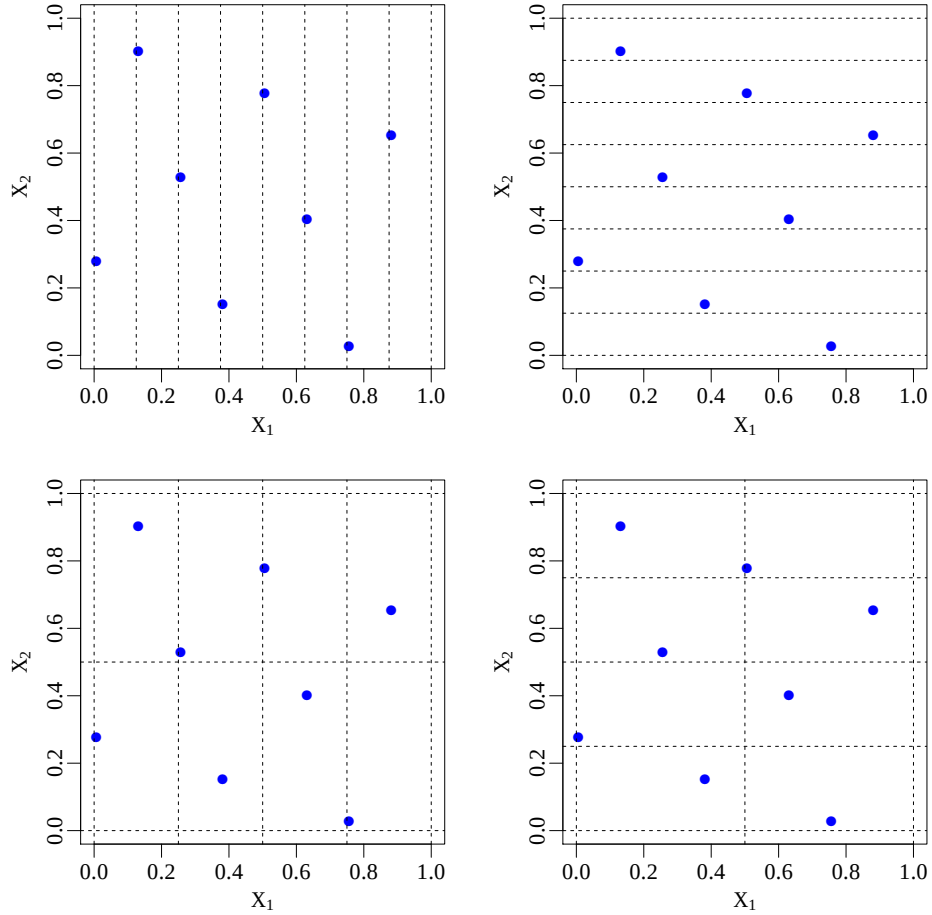
La construction de C_j requiert un polynôme primitif dans $GF(2)$ et une suite de nombres appelés nombres directionnels. Notons par $p_j(z)$ le polynôme primitif utilisé :

$$p_j(z) = z^{d_j} + a_{1,j}z^{d_j-1} + \dots + a_{d_j,j},$$

où d_j désigne le degré du polynôme et $\forall k \in \{1, \dots, d_j\}$, $a_{k,j} \in GF(2)$. Les nombres directionnels sont définis à partir d'une suite d'entiers $\{m_{1,j}, m_{2,j}, \dots\}$ construite par la récurrence suivante :

$$m_{k,j} = 2a_{1,j}m_{k-1,j} \oplus 2^2 a_{2,j}m_{k-2,j} \oplus \dots \oplus 2^{d_j-1} a_{d_j-1,j}m_{k-d_j+1,j} \oplus 2^{d_j} m_{k-d_j,j} \oplus m_{k-d_j,j}$$

FIGURE 1.7 – Les 4 partitions possibles du carré $[0, 1]^2$ pour un $(0, 3, 2)$ -réseau en base 2. Les couples (γ_1, γ_2) engendrant ces partitions sont : $(3, 0)$ (haut-gauche), $(0, 3)$ (haut-droite), $(2, 1)$ (bas-gauche), $(1, 2)$ (bas-droite). Chacun des 8 intervalles élémentaires contient exactement 1 point.



où $\oplus$ désigne l'addition dans $GF(2)$. Les d_j premières valeurs $m_{1,j}, \dots, m_{d_j,j}$ sont choisies librement sous la condition que chaque terme $m_{k,j}$, $1 \leq k \leq d_j$, soit impair et vérifie $m_{k,j} < 2^k$. Les nombres directionnels $\{v_{1,j}, v_{2,j}, \dots\}$ sont alors définis par :

$$v_{k,j} = \frac{m_{k,j}}{2^k}, \quad v_{k,j} = \sum_{l=1}^{\infty} v_{k,j,l} 2^{-l}, \quad v_{k,j,l} \in \{0, 1\}.$$

et la matrice C_j s'écrit :

$$C_j = \begin{pmatrix} 1 & v_{2,j,1} & v_{3,j,1} & \dots \\ 0 & 1 & v_{3,j,2} & \dots \\ 0 & 0 & 1 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Les matrices C_j sont de taille infinie puisqu'elles sont issues d'une construction récursive. Nous décrivons maintenant la construction d'une suite de Sobol' :

Définition 1.10. Soient $C_1, \dots, C_d$ un ensemble de d matrices génératrices dont on note par $C_1^m, \dots, C_d^m$ les blocs supérieurs gauche de taille $m \times m$. Notons par $\{\mathbf{X}_i\}_{i=0}^{2^m-1}$ les 2^m premiers points d'une suite de Sobol'. Pour $i = 0, \dots, 2^m - 1$, le point $\mathbf{X}_i = (\phi_2(X_{i,1}), \dots, \phi_2(X_{i,d}))$ est défini par :

$$\begin{aligned} \phi_2(X_{i,j}) &= \sum_{k=1}^m x_{i,j,k} 2^{-k}, \quad j = 1, \dots, d, \\ (x_{i,j,1}, \dots, x_{i,j,m})^\top &= C_j^m \mathbf{i}, \end{aligned} \quad (1.10)$$

où $\mathbf{i} = (i_0, \dots, i_{m-1})^\top$ est le vecteur résultant de la décomposition binaire (1.7) de i .

Les opérations matricielles ont lieu dans $GF(2)$. Pour construire les 2^m prochains points $\mathbf{X}_{2^m}, \dots, \mathbf{X}_{2^{m+1}-1}$ de la suite, on déduit de (1.10) la récurrence suivante :

$$(x_{i+2^m,j,1}, \dots, x_{i+2^m,j,m+1})^\top = (x_{i,j,1}, \dots, x_{i,j,m}, 0)^\top \oplus (c_j^{m+1})^\top, \quad i = 0, \dots, 2^m - 1,$$

où c_j^{m+1} est la dernière colonne de C_j^{m+1} .

Nous énonçons maintenant quelques propriétés des suites de Sobol' relatives à la t -valeur.

Proposition 1.3. Soit une suite de Sobol' définie par (1.10). Toute projection d'ordre un de la suite est une $(0, 1)$ -suite digitale.

Démonstration. Nous renvoyons à la remarque 3.5 dans [Sob67]. □

Cette proposition nous fournit la justification de la bonne répartition des points d'une suite de Sobol' dans les projections d'ordre 1. Le choix du polynôme primitif $p_j(z)$ pour construire chaque matrice C_j a un impact sur la t -valeur de la suite :

Théorème 1.3. Une suite de Sobol' de dimension d définie par d polynômes primitifs de degré respectif $d_1, \dots, d_d$ est une (t, d) -suite où :

$$t = \sum_{j=1}^d (d_j - 1).$$

Démonstration. Nous renvoyons à la preuve du théorème 3.3 dans [Sob67]. □

D'après ce théorème, il est important de ranger les polynômes primitifs utilisés par ordre croissant de leur degré pour obtenir une t -valeur la plus petite possible. Si c'est le cas $t \in O(d \log(d))$. Dans [JK08], Joe et Kuo proposent un choix des matrices C_j optimisant les t -valeurs des projections d'ordre 2 d'une suite de Sobol'.

Le choix des nombres directionnels peut avoir un effet sur des sous-ensembles finis de points d'une suite de Sobol'. Dans [Sob67], Sobol' propose un choix de ces

nombre afin de satisfaire des propriétés d'équidistribution (appelées propriétés A et A' dans [Sob67]) des points de la suite.

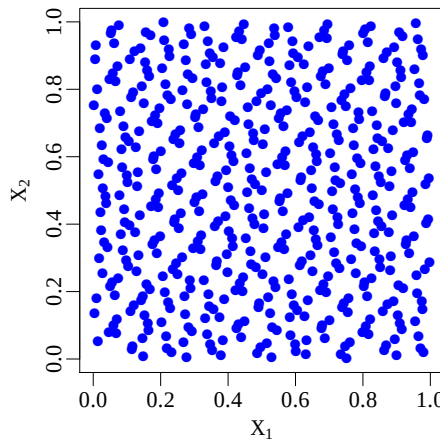
D'autres suites digitales ont été construites ayant pour objectif d'améliorer les résultats obtenus par Sobol' sur le critère t -valeur. Dans [Fau82], Faure propose de construire des $(0, d)$ -suites en base b où b est un nombre premier. Dans [Nie92b], Niederreiter étend les résultats de Faure pour une base b puissance d'un nombre premier. On signalera toutefois que dans les deux cas, la dimension d de la suite doit satisfaire $d \leq b$. Niederreiter propose également dans ses travaux avec Xing [NX96] des constructions de (t, d) -suites avec de meilleures bornes sur la t -valeur que celle obtenue par Sobol'. Cependant la construction de ces suites est plus complexe.

1.3.3 Méthodes de randomisation

Pour une fonction f , la vitesse de convergence de sa moyenne μ estimée par une (t, d) -suite en base b est en $O(\log^d(n)/n)$ si n est de la forme $n = b^m$. Ce résultat provient de l'inégalité de Koksma-Hlawka [Hla61] et est valable uniquement sous la condition que f soit à variation bornée au sens de Hardy et Krause [Nie92b]. Par conséquent, pour d grand, la taille n du plan requise pour que ce taux de convergence soit comparable à celui obtenu par un échantillonnage Monte Carlo devient rapidement trop élevée.

De plus, les points des suites digitales forment des structures géométriques dans $[0, 1]^d$ issues de leur construction algébrique. Ces structures ont tendance à altérer l'estimation de μ . La figure 1.8 illustre un exemple de ces structures pour les 512 premiers points d'une suite de Sobol' en dimension 2.

FIGURE 1.8 – 512 premiers points d'une suite de Sobol' en dimension 2.



Une solution possible à ce dernier problème consiste à faire appel à des méthodes de randomisation. Ces méthodes cherchent à ce que chaque point $\mathbf{x}_i$ de la suite randomisée soit uniformément distribué sur $[0, 1]^d$ tout en conservant la structure

de (t, d) -suite. Nous présentons dans cette section quelques-unes de ces méthodes en commençant par la plus simple qui consiste à appliquer un *random shift*.

Un *random shift*, encore appelé rotation de Cranley-Patterson [CP76], consiste à ajouter à chaque variable X_j , $j \in \mathcal{D}$, de la suite une réalisation d'une variable aléatoire uniformément distribuée sur $[0, 1]$. Soit $\{\mathbf{X}_i\}_{i \in \mathbb{N}}$ une suite digitale, la suite randomisée associée $\{\mathbf{Z}_i\}_{i \in \mathbb{N}}$ est obtenue par :

$$\mathbf{Z}_i = (X_{i,1} + U_1 \mod 1, \dots, X_{i,d} + U_d \mod 1), \quad i \in \{1, 2, \dots\},$$

où $\mod$ désigne l'opérateur modulo et $U_1, \dots, U_d$ sont d variables aléatoires uniformément distribuées sur $[0, 1]$. Pour $i \in \{1, \dots, n\}$, $\mathbf{Z}_i$ est uniformément distribuée dans $[0, 1]^d$. Toutefois, un *random shift* ne préserve pas nécessairement la structure de (t, d) -suite.

Une alternative consiste à utiliser des méthodes de *scrambling*. Ces méthodes travaillent en profondeur, sur la structure digitale même de la suite. La méthode originelle est due à Owen et consiste à appliquer des permutations sur les coefficients $x_{i,j,k}$ issus de la décomposition en base b (1.8) des coordonnées d'un point $\mathbf{X}_i$ de la suite digitale. La permutation appliquée à $x_{i,j,k}$ dépend de celles appliquées aux coefficients précédents $x_{i,j,h}$ pour $h < k$. Pour plus de précision sur cette méthode de randomisation, nous renvoyons le lecteur à son introduction dans [Owe95, section 3]. La suite randomisée $\{\mathbf{Z}_i\}_{i \in \mathbb{N}}$ issue du *scrambling* proposé par Owen hérite des propriétés de la suite dont elle est issue :

Proposition 1.4. *Si $\{\mathbf{X}_i\}_{i \in \mathbb{N}}$ est une (t, d) -suite en base b , alors la suite randomisée $\{\mathbf{Z}_i\}_{i \in \mathbb{N}}$ est également une (t, d) -suite en base b .*

Démonstration. Nous renvoyons à la preuve de la proposition 1 dans [Owe95]. $\square$

De plus :

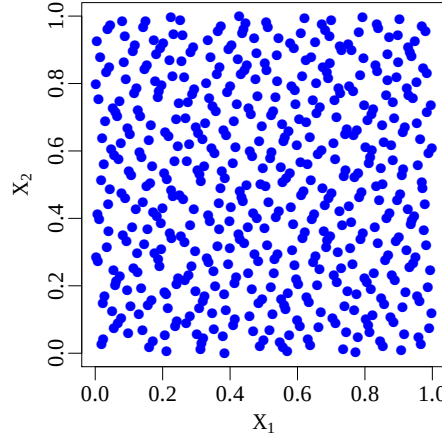
Proposition 1.5. *Soit la suite randomisée $\{\mathbf{Z}_i\}_{i \in \mathbb{N}}$ obtenue suivant le *scrambling* d'Owen, alors pour chaque i , $\mathbf{Z}_i$ est uniformément distribuée dans $[0, 1]^d$.*

Démonstration. Nous renvoyons à la preuve de la proposition 2 dans [Owe95]. $\square$

La méthode de *scrambling* proposée par Owen peut être trouvée dans la littérature sous le nom de *fully random scrambling* ou encore *nested uniform scrambling*. Cependant, cette méthode est coûteuse d'un point de vue computationnel car elle demande de garder en mémoire une grande quantité de nombres issus des permutations ($O(b^k)$ nombres pour les k premiers coefficients et ce pour chaque coordonnée j). La figure 1.9 illustre les 512 premiers points de la suite de Sobol' de la figure 1.8, randomisée par un *fully random scrambling*.

Une alternative moins coûteuse ($O(k)$ au lieu de $O(b^k)$) que la méthode d'Owen a été étudiée dans [Mat98, HH03]. Cette alternative est appelée *random linear*

FIGURE 1.9 – 512 points de la suite de Sobol' illustrée en figure 1.8 et randomisée par le *scrambling* d'Owen.



scrambling. Elle consiste dans un premier temps à choisir d matrices triangulaires inférieures de rang plein, notées $L_1, \dots, L_d$, construites sur $GF(b)$, puis à multiplier les matrices génératrices $C_1, \dots, C_d$ de la (t, d) -suite à gauche par $L_1, \dots, L_d$. Ainsi, la suite randomisée résultante est une suite digitale de matrices génératrices $L_1 C_1, \dots, L_d C_d$. Dans [Mat99], Matousek montre que la t -valeur de la suite randomisée obtenue n'est pas plus large que celle de la suite d'origine. Cette méthode de randomisation assure également que chaque point randomisé soit uniformément distribué dans $[0, 1]^d$ (Remarque 3.2 dans [Owe03]).

Il existe encore d'autres méthodes de *scrambling* que nous choisissons de ne pas détailler, le lecteur intéressé peut trouver une synthèse de ces méthodes dans [Lem09, section 6.2].

Nous concluons cette section par des propriétés relatives aux suites obtenues par les méthodes de *scrambling* présentées ci-dessus :

Proposition 1.6. *Soit une fonction $f \in \mathbb{L}^2([0, 1]^d)$ de moyenne μ et variance $\sigma^2 < +\infty$. Notons par $\hat{\mu}_{scr}$ l'estimation de μ par un (t, m, d) -réseau randomisé par un *fully random scrambling*. Soit $n = b^m$, alors :*

$$\text{Var}[\hat{\mu}_{scr}] \leq \frac{b^t}{n} \left(\frac{b+1}{b-1} \right)^d \sigma^2.$$

Démonstration. Nous renvoyons à la preuve du théorème 1 dans [Owe98]. $\square$

Le théorème suivant est spécifique au cas de fonctions lisses :

Théorème 1.4. *Si f est une fonction lisse, c'est à dire s'il existe $C \geq 0$ et $\beta \in]0, 1]$ tel que :*

$$\left| \frac{\partial^d}{\partial x_1 \dots \partial x_d} f(\mathbf{x}) - \frac{\partial^d}{\partial x'_1 \dots \partial x'_d} f(\mathbf{x}') \right| \leq C \|\mathbf{x} - \mathbf{x}'\|^\beta, \quad \forall (\mathbf{x}, \mathbf{x}') \in ([0, 1]^d)^2,$$

où $\|\cdot\|$ désigne la norme euclidienne, alors pour un réseau digital randomisé par un *fully random scrambling* ou un *random linear scrambling* :

$$\text{Var}[\hat{\mu}_{scr}] \in O(n^{-3} \log^{d-1}(n)).$$

Démonstration. Nous renvoyons à la preuve du théorème 2 dans [Owe97b]. $\square$

Ces deux résultats montrent que randomiser une suite digitale permet potentiellement d'améliorer l'estimation de la moyenne μ . Toutefois, comme les suites digitales randomisées sont des (t, d) -suites, dans le pire des cas la vitesse de convergence reste en $O(\log^d(n)/n)$.

Cet ordre de convergence est cependant pessimiste car la vitesse de convergence d'une (t, d) -suite est plus ou moins liée à la dimension effective de la fonction f . La dimension effective d'une fonction f se définit à partir de la décomposition ANOVA fonctionnelle de f :

$$f(\mathbf{x}) = \sum_{u \subseteq \mathcal{D}} f_u(\mathbf{x}),$$

détaillée plus précisément à la section 2.1.2. Notons par σ^2 et σ_u^2 les variances respectives de f et f_u . Etant donné une proportion p , la dimension effective de f au sens de superposition est le plus petit entier d_S tel que :

$$\frac{1}{\sigma^2} \sum_{0 \leq \#u \leq d_S} \sigma_u^2 \geq p.$$

Autrement dit la fonction f peut être approchée par une somme de fonctions d'au plus d_S variables. La dimension effective de f au sens de troncation est le plus petit entier d_T tel que :

$$\frac{1}{\sigma^2} \sum_{u \subseteq \{1, 2, \dots, d_T\}} \sigma_u^2 \geq p.$$

Autrement dit la fonction f peut être approchée par une somme de fonctions définies seulement à partir des d_T premières variables. Pour plus de précisions sur la notion de dimension effective nous renvoyons le lecteur à l'article de Caflisch *et al.* [CMO97].

Dans [WF03], les auteurs remarquent, illustrations à l'appui, que dans le cas de fonctions ayant une petite dimension effective au sens de troncation (≤ 10) ou de superposition (≤ 3), on peut s'attendre à ce que la vitesse de convergence d'une (t, d) -suite soit meilleure que celle d'un plan aléatoire.

1.4 Critères de qualité

La dernière section de ce chapitre est dédiée à la présentation de critères permettant d'étudier les propriétés d'uniformité ou de bon remplissage de l'espace (*space-filling*) d'un plan d'expériences numériques. Les critères que nous présentons dans cette section ne requièrent aucune hypothèse supplémentaire sur le modèle f étudié.

Le premier critère est appelé discrédance et permet de mesurer si la distribution des points d'un plan d'expériences est proche ou non d'une distribution uniforme.

Le second critère que nous présentons est un critère de distance entre points permettant d'évaluer la qualité de remplissage de l'espace d'un plan d'expériences.

Le dernier critère s'inspire de la théorie des graphes, ce critère permet de classer la distribution des points d'un plan d'expériences parmi différentes catégories de distributions.

Combiner l'utilisation de ces différents critères permet d'investiguer avec précision la qualité d'un plan d'expériences à partir de ses propriétés d'uniformité et de remplissage de l'espace.

1.4.1 Discrédance

La discrédance permet de juger l'uniformité d'un plan d'expériences. Ce critère peut être vu comme une généralisation multidimensionnelle de la statistique de Kolmogorov-Smirnov. Afin de définir ce critère nous introduisons l'ensemble des parties de $[0, 1]^d$ de la forme :

$$[0, \mathbf{v}] = \left\{ \mathbf{X} \in [0, 1]^d \mid 0 \leq X_j \leq v_j, \forall j \in \mathcal{D} \right\}, \quad \mathbf{v} = (v_1, \dots, v_d) \in [0, 1]^d.$$

Ces ensembles sont des pavés dont l'un des sommets est l'origine $\mathbf{0}$ de $[0, 1]^d$. Pour un plan d'expériences $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$, nous comptons combien de points du plan $\mathcal{P}$ appartiennent à chaque pavé. C'est à dire que nous déterminons la cardinalité suivante :

$$\alpha(\mathcal{P}, \mathbf{v}) = \# \{ \mathbf{X}_i \in \mathcal{P} \cap [0, \mathbf{v}] \},$$

où $\#$ désigne l'opérateur cardinalité. La fonction de répartition empirique de $\mathcal{P}$ assigne une probabilité de $\alpha(\mathcal{P}, \mathbf{v})/n$ au pavé $[0, \mathbf{v}]$. Pour une loi uniforme sur $[0, 1]^d$, cette probabilité vaut $\prod_{j=1}^d v_j$. On peut alors mesurer l'écart, la discrédance, entre la distribution des points de $\mathcal{P}$ et une loi uniforme, par :

$$D^*(\mathcal{P}) = \sup_{\mathbf{v} \in [0, 1]^d} \left| \prod_{j=1}^d v_j - \alpha(\mathcal{P}, \mathbf{v})/n \right|. \quad (1.11)$$

$D^*(\mathcal{P})$ est appelée la discrédance étoilée d'un ensemble de points $\mathcal{P}$. Une variante de cette mesure est la discrédance extrême qui ne restreint pas un des sommets des pavés à être à l'origine. Les pavés étudiés sont alors définis par :

$$[\mathbf{w}, \mathbf{v}] = \left\{ \mathbf{X} \in [0, 1]^d \mid w_j \leq X_j \leq v_j, \forall j \in \mathcal{D} \right\}, \quad (\mathbf{v}, \mathbf{w}) \in ([0, 1]^d)^2,$$

à partir desquels on dénombre :

$$\alpha(\mathcal{P}, \mathbf{w}, \mathbf{v}) = \# \{ \mathbf{X}_i \in \mathcal{P} \cap [\mathbf{w}, \mathbf{v}] \}.$$

La discrédance extrême est alors définie par :

$$D(\mathcal{P}) = \sup_{(\mathbf{w}, \mathbf{v}) \in ([0,1]^d)^2} \left| \prod_{j=1}^d (v_j - w_j) - \alpha(\mathcal{P}, \mathbf{w}, \mathbf{v})/n \right|. \quad (1.12)$$

Pour les dimensions $d = 1$ et $d = 2$ des formules analytiques ont été trouvées pour (1.11) et (1.12) [Nie92b]. Malheureusement, le calcul de ces quantités devient trop complexe pour des dimensions plus grandes. Les expressions (1.11) et (1.12) font intervenir une norme $\mathbb{L}^\infty$, une alternative consiste à considérer une norme $\mathbb{L}^2$. On définit alors la discrédance $\mathbb{L}^2$ étoilée et la discrédance $\mathbb{L}^2$ extrême par :

$$D_2^*(\mathcal{P}) = \left(\int_{[0,1]^d} \left(\prod_{j=1}^d v_j - \alpha(\mathcal{P}, \mathbf{v})/n \right)^2 d\mathbf{v} \right)^{1/2}, \quad (1.13)$$

$$D_2(\mathcal{P}) = \left(\int_{[0,1]^{2d}} \left(\prod_{j=1}^d (v_j - w_j) - \alpha(\mathcal{P}, \mathbf{w}, \mathbf{v})/n \right)^2 d\mathbf{w} d\mathbf{v} \right)^{1/2}. \quad (1.14)$$

Des expressions analytiques de (1.13) et (1.14) peuvent être obtenues [Hic98a]. Pour un plan aléatoire $\mathcal{P}$ à n points, nous avons les résultats suivants :

$$\begin{aligned} \mathbb{E}[D_2^*(\mathcal{P})^2] &= (2^{-d} - 3^{-d})/n, \\ \mathbb{E}[D_2(\mathcal{P})^2] &= 6^{-d}(1 - 2^{-d})/n. \end{aligned}$$

Les discrédances $\mathbb{L}^2$ sont les plus utilisées en pratique du fait de leurs expressions analytiques simples et rapides à évaluer. Pour des petites valeurs de n , la discrédance $\mathbb{L}^2$ extrême n'est pas assez précise (approximations grossières observées sur des cas tests) tandis que la discrédance $\mathbb{L}^2$ étoilée ne capte pas la différence $\prod_{j=1}^d v_j - \alpha(\mathcal{P}, \mathbf{v})/n$ dans des variétés géométriques (*manifolds*) de faible dimension.

Pour remédier à ces défauts, Hickernell propose plusieurs versions modifiées de la discrédance $\mathbb{L}^2$ étoilée [Hic98a, Hic98b, FLS06] parmi lesquelles les discrédances $\mathbb{L}^2$ centrée et $\mathbb{L}^2$ enveloppée. Les formules analytiques de ces diverses discrédances sont explicitées dans [Fra08].

Pour l'étude des propriétés d'uniformité de nos plans construits aux chapitres 3 à 5, on se limite à l'utilisation de la discrédance $\mathbb{L}^2$ étoilée. En effet, dans nos exemples d'application, les tailles n des plans construits sont suffisamment grandes pour pallier les défauts de cette discrédance.

1.4.2 Distance entre points

Dans [JMY90], Johnson introduit deux critères de distance entre points. Ces critères permettent d'évaluer si la distribution des points d'un plan d'expériences

$\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ est proche de celle d'une grille régulière ou non. Le premier critère, appelé critère *minimax*, est défini par :

$$\phi_{mM}^p(\mathcal{P}) = \max_{\mathbf{v} \in [0,1]^d} \min_{\mathbf{X}_i \in \mathcal{P}} \|\mathbf{v} - \mathbf{X}_i\|_p, \quad (1.15)$$

où $\|\cdot\|_p$ désigne la norme $\mathbb{L}^p$ dans $[0,1]^d$:

$$\|\mathbf{v}\|_p = \left(\sum_{j=1}^d v_j^p \right)^{1/p}.$$

Le critère *minimax* est appelé critère de dispersion par Niederreiter [Nie92b, Chapitre 6]. Ce critère est le plus couramment défini pour $p = 2$ et l'on abrège le critère par ϕ_{mM} . Une petite valeur de $\phi_{mM}(\mathcal{P})$ signifie qu'il n'existe pas de point $\mathbf{v}$ dans $[0,1]^d$ qui ne soit pas proche d'un point $\mathbf{X}_i$ de l'ensemble $\mathcal{P}$. L'utilisation de ce critère requiert le calcul de toutes les distances entre les points $\mathbf{v}$ du domaine $[0,1]^d$ et les points $\mathbf{X}_i$ de $\mathcal{P}$. En pratique, ce critère est approximé à l'aide d'une discrétisation fine du domaine $[0,1]^d$, les points issus de la discrétisation substituant $\mathbf{v}$ dans (1.15). Dans [PM12], Pronzato et Müller propose un calcul exact de $\phi_{mM}(\mathcal{P})$ à partir de la triangulation de Delauney de l'ensemble de points $\mathcal{P}$. Ces deux dernières approches (approximation et calcul exact) ne sont cependant envisageables que pour des petites valeurs de d .

Le second critère, appelé critère *maximin*, est défini par :

$$\phi_{Mm}^p(\mathcal{P}) = \min_{(\mathbf{X}_i, \mathbf{X}_j) \in \mathcal{P}^2} \|\mathbf{X}_i - \mathbf{X}_j\|_p. \quad (1.16)$$

Le critère *maximin* est également le plus souvent défini pour $p = 2$ et l'on abrège ce critère par ϕ_{Mm} . Ce critère ne fait appel qu'aux points du plan $\mathcal{P}$ et évalue le minimum des distances entre toutes les paires de points de $\mathcal{P}$. Si ce minimum vaut c , tous les points de $\mathcal{P}$ sont espacés les uns des autres d'une distance d'au moins c . Ainsi, plus la valeur de ϕ_{Mm} est grande, meilleur est le remplissage de l'espace par $\mathcal{P}$.

1.4.3 Arbre de poids minimal

L'introduction du critère arbre de poids minimal [FVCS09], encore appelé *emst* (*euclidean minimal spanning tree*), nécessite au préalable quelques rappels sur la notion de graphe.

Un graphe est un ensemble de points nommés noeuds (sommets) reliés par des arêtes. Formellement, on note l'ensemble des noeuds du graphe par $V = \{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ et l'ensemble des arêtes par $E = \{e = (\mathbf{v}, \mathbf{v}') | (\mathbf{v}, \mathbf{v}') \in V^2\}$. Dans notre cas, les arêtes sont non orientées. Le graphe associé est noté $G = (V, E)$.

Dans le cas général, un graphe peut posséder des arêtes dites multiples qui relient toutes la même paire de sommets. Un graphe peut également posséder des boucles, c'est à dire des sommets reliés à eux-mêmes. Nous introduisons maintenant quelques définitions relatives aux graphes :

- un cycle est une suite d'arêtes de E reliant un sommet à lui même.
- un graphe pondéré sur ces arêtes est un graphe auquel on associe une fonction de pondération θ de ses arêtes. C'est à dire que chaque arête $e = (v, v')$ possède un poids $\theta(e)$.
- un graphe simple est un graphe ne possédant ni arêtes multiples ni boucles.
- un graphe connexe est un graphe tel que : quelque soit la paire de sommets distincts $(v, v') \in V^2$ considérée, il existe une suite d'arêtes de E reliant v à v' .

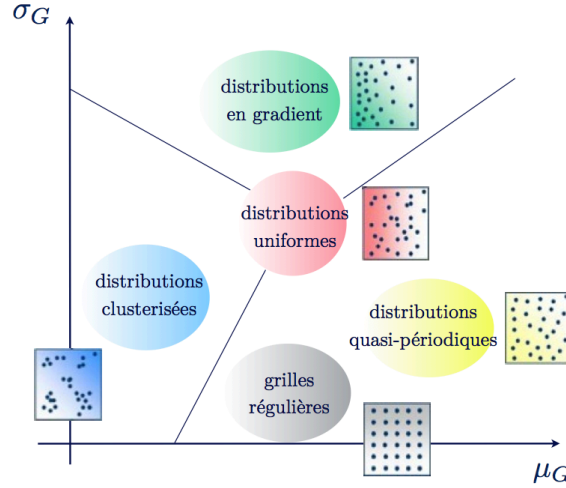
Un graphe est un arbre s'il est simple, connexe et acyclique (ne possède pas de cycles). Un graphe est un arbre de poids minimal s'il est un arbre dont la somme des poids de ses arêtes est minimale.

Le critère *emst* pour un plan $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ consiste à construire un arbre de poids minimal $G = (V, E)$ où l'ensemble des sommets V s'identifie aux points du plan $\mathcal{P}$. Le poids de chaque arête $e \in E$ correspond à la valeur de la distance euclidienne entre les sommets reliés par e : $\theta(e) = \|\mathbf{X}_i - \mathbf{X}_k\|_2$, $(\mathbf{X}_i, \mathbf{X}_k) \in \mathcal{P}^2$.

Pour un même ensemble de points, on peut construire plusieurs arbres de poids minimal. Cependant, l'histogramme des poids des arêtes est unique [FVCS09]. Le plan $\mathcal{P}$ peut ainsi être caractérisé par la moyenne μ_G et l'écart-type σ_G des poids des arêtes de l'arbre G . Le critère *emst* peut être représenté dans un graphe d'interprétation (μ_G, σ_G) , illustré en figure 1.10, afin de comparer la distribution des points de $\mathcal{P}$ à différentes catégories de distributions ; la distribution uniforme faisant office de distribution de référence.

Parmi ces catégories, les distributions quasi-périodiques sont caractérisées par une grande valeur de μ_G et une faible valeur de σ_G . Une grande valeur de μ_G signifie qu'en moyenne les points sont bien espacés les uns des autres. Une faible valeur de σ_G que les distances entre les points sont homogènes. Ainsi les distributions quasi-périodiques remplissent efficacement le domaine $[0, 1]^d$ du point de vue de la distance entre points.

Il est possible de définir le critère de comparaison suivant : un plan $\mathcal{P}_1$ remplit mieux l'espace qu'un plan $\mathcal{P}_2$, si $\mu_{G_1} > \mu_{G_2}$ et $\sigma_{G_1} < \sigma_{G_2}$ avec G_1 et G_2 les arbres respectifs de $\mathcal{P}_1, \mathcal{P}_2$. Le critère *emst* est un bon complément du critère *maximin* pour diagnostiquer avec précision les propriétés *space-filling* d'un plan d'expériences. Il est également possible de construire d'autres types d'arbres (par exemple l'arbre des plus proches voisins [PM12]) pour évaluer les propriétés *space-filling* d'un plan d'expériences.

FIGURE 1.10 – Graphe d'interprétation (μ_G, σ_G) du critère *emst* pour un arbre G .

1.5 Synthèse

Nous avons présenté dans ce chapitre deux classes de plans numériques, les plans stratifiés et les suites digitales (et réseaux digitaux). Les plans stratifiés assurent une stratification raffinée des variables d'entrée dans certaines projections. Les suites digitales fournissent une bonne répartition des points de chaque variable d'entrée et la répartition des points dans $[0, 1]^d$ est caractérisée par la t -valeur de la suite. Pour ces deux classes de plans, nous avons souligné l'importance d'une randomisation des points du plan d'expériences. Cette randomisation a pour premier objectif de détruire les structures géométriques issues des constructions algébriques pouvant donner lieu à des corrélations indésirables. Le second objectif est d'obtenir une distribution plus uniforme des points tout en conservant les bonnes propriétés de remplissage de l'espace du plan.

Nous avons conclu ce chapitre par la présentation de critères permettant de mesurer la qualité des propriétés d'uniformité et de remplissage de l'espace d'un plan d'expériences. Nous avons seulement décrit ces critères comme des outils d'évaluation des propriétés d'un plan. Cependant, ces critères peuvent également être utilisés comme des critères d'optimisation dans des algorithmes visant à construire des plans optimaux [DCI13].

Nous avons volontairement omis la présentation de tels plans dont la construction est pertinente uniquement pour un petit nombre de points, de l'ordre de 10^2 à 10^3 . Pour des tailles de plans plus grandes, ces algorithmes deviennent coûteux en temps de calcul et n'offrent aucune garantie quant à l'obtention d'un optimum. Dans ce cas, une alternative plus efficace consiste à construire plusieurs plans et conserver celui obtenant le meilleur score pour le critère sélectionné.

Nous avons également omis de mentionner les constructions dites récursives de

plans d'expériences. Les constructions récursives pertinentes pour nos travaux de thèse sont détaillées dans les chapitres 4 et 5 portant sur la question d'estimation récursive des indices de Sobol'.

Pour chacun des plans présentés nous nous sommes attachés à donner des résultats relatifs à la variance de l'estimation de la moyenne μ d'une fonction f . De prime abord, de tels résultats ne semblent présenter que peu d'intérêt dans le cadre de l'analyse de sensibilité. Nous montrerons pourtant dans le prochain chapitre que les indices de Sobol' peuvent se réécrire comme une expression de plusieurs moyennes (voir équation (2.10)). Ainsi, l'estimation des indices de Sobol' peut potentiellement être améliorée par l'utilisation de plans offrant de meilleurs résultats de convergence pour l'estimation de ces moyennes.

Dans nos travaux présentés aux chapitres 3 à 6, les plans d'expériences construits reposent sur les différentes structures présentées dans ce chapitre. Ces structures sont modifiées pour pouvoir faire face aux problématiques énoncées dans l'introduction (présence de corrélation, sorties multivariées, estimation récursive).

CHAPITRE 2

Estimation des indices de Sobol' : méthodes Monte Carlo

Comme énoncé en introduction, les indices de Sobol' sont des indicateurs de sensibilité permettant de quantifier l'influence d'une variable d'entrée ou d'un groupe de variables d'entrée sur une sortie d'un modèle numérique. Ce chapitre est consacré à la présentation de méthodes Monte Carlo pour l'estimation des indices de Sobol'. La première partie 2.1 porte sur la définition des indices de Sobol'. La partie 2.2 s'intéresse aux méthodes d'estimation de ces indices issues de l'approche Monte Carlo originelle proposée par Sobol' [Sob93]. Nous présentons ensuite dans la partie 2.3 des méthodes reposant sur l'utilisation de plans répliqués.

Ce chapitre ne traite pas des méthodes d'estimation basées sur l'analyse harmonique (méthodes FAST [CFS⁺73, CSS75, CLS78], RBD et RBD-FAST [TGM06]) ni des méthodes d'estimation reposant sur une décomposition ANOVA spectrale de la sortie étudiée (par exemple décomposition par polynômes de chaos [GS03]). Ce chapitre ne considère pas non plus les méthodes d'estimation basées sur l'utilisation d'un metamodelle. Pour de telles méthodes, nous renvoyons le lecteur aux travaux de thèse de Janon [Jan08].

Nous conservons pour ce chapitre les mêmes notations que celles introduites dans le chapitre 1 (page 9). A cette occasion, nous rappelons que l'on suppose $f \in \mathbb{L}^2([0, 1]^d, \lambda(d\mathbf{x}))$ et $Y = f(\mathbf{X})$, $\mathbf{X} = (X_1, \dots, X_d)$ étant un vecteur aléatoire uniformément distribué sur $[0, 1]^d$.

L'incertitude sur les entrées est modélisée par $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,d})$, $i \in \mathbb{N}$. Soit $u \subset \mathcal{D}$, $\subset$ désignant l'inclusion stricte, nous notons par $-u$ son complémentaire et par $\#u$ sa cardinalité. $\mathbf{X}_u$ (respectivement $\mathbf{X}_{i,u}$) désigne le vecteur de composantes X_j (respectivement $X_{i,j}$) pour $j \in u$.

Soit $\mathbf{Z} = (Z_1, \dots, Z_d)$ un vecteur identiquement distribué suivant la loi de $\mathbf{X}$ et indépendant de $\mathbf{X}$. On désigne par $\mathbf{X}_u : \mathbf{Z}_{-u}$ le vecteur dont les composantes d'indices $j \in u$ sont les X_j et dont les autres composantes sont les Z_j . Pour $i \in \mathbb{N}$, $\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}$ est appelé point hybride.

2.1 Indices de sensibilité

Cette section a pour objectif de contextualiser la définition des indices de Sobol'. Pour cela nous repartons de la définition d'une statistique appelée mesure d'import-

tance [SCS08]. Puis, nous introduisons la décomposition dite ANOVA fonctionnelle dont est issue la définition des indices de Sobol'.

2.1.1 Mesure d'importance

Il est naturel de mesurer l'importance d'une variable X_j sur une quantité d'intérêt Y par la mesure suivante :

$$\frac{\text{Var}_{X_j} [\text{E}[Y|X_j = x_j]]}{\text{Var}[Y]}, \quad (2.1)$$

où $\text{E}[Y|X_j = x_j]$ désigne l'espérance conditionnelle de Y sachant $X_j = x_j$ et la notation Var_{X_j} désigne la variance lorsque X_j varie sur l'ensemble de son domaine.

Cette mesure d'importance est décrite par différents termes dans la littérature. Dans [McK95], McKay appelle le numérateur de (2.1) *variance correlation expectation* (abrégée par VCE) et qualifie le quotient de *correlation ratio*. A l'origine, cette mesure d'importance a été introduite comme un indicateur, noté η^2 , de liaison entre une variable aléatoire quantitative et une variable aléatoire qualitative. Des synthèses exhaustives de cette mesure d'importance sont proposées dans [KS67] et [Sap75].

Dans [IH90a], Iman et Hora montrent que cette mesure d'importance n'est pas robuste et qu'elle peut être influencée par la présence de valeurs aberrantes (*outliers*) dans le cas de distributions à queue lourde (par exemple une loi log-normale). Iman et Hora proposent alors plusieurs versions alternatives de cette mesure d'importance. Dans [HI86], ils considèrent la quantité suivante mesurant l'importance d'une variable X_j :

$$I_j = \sqrt{\text{Var}[Y] - \text{E} [\text{Var}_{X_j}[Y|X_j = x_j]]}.$$

Le théorème de la variance totale nous donne :

$$\text{Var}[Y] = \text{Var}_{X_j} [\text{E}[Y|X_j = x_j]] + \text{E}_{X_j} [\text{Var}[Y|X_j = x_j]],$$

et l'on récupère ainsi l'expression du numérateur de (2.1). Toutefois, il est montré que la mesure I_j n'est pas robuste et est également influencée par la présence d'*outliers*.

Dans [IH90a], Iman et Hora proposent ensuite une nouvelle mesure définie par :

$$\frac{\text{Var}_{X_j} [\text{E}[\log(Y)|X_j = x_j]]}{\text{Var}[\log(Y)]},$$

où $\text{E}[\log(Y)|X_j = x_j]$ est estimée par une régression linéaire. Cette mesure est démontrée comme étant plus robuste que la précédente. Cependant il est difficile d'interpréter les conclusions obtenues avec $\log(Y)$ par rapport à la sortie d'origine Y du modèle.

Les indices de Sobol' dits d'ordre un s'identifient à la mesure d'importance définie par l'équation (2.1). La construction des indices de Sobol' repose sur une décomposition, appelée décomposition ANOVA, du modèle f étudié.

2.1.2 Décomposition ANOVA

Une synthèse concise de la décomposition ANOVA est proposée dans [Tis12]. Ici, nous rappelons la décomposition ANOVA d'un modèle f dans le cadre de variables aléatoires indépendantes. Cette décomposition est issue des travaux initiaux de Hoeffding [Hoe48], et a été formalisée par Efron et Stein [Efr81]. Dans [Sob93], Sobol' introduit cette décomposition de la manière suivante :

Définition 2.1. *Soit $f \in \mathbb{L}^2([0,1]^d)$, le modèle f admet une décomposition en une somme de termes de dimensions croissantes s'écrivant :*

$$f(\mathbf{x}) = f_0 + \sum_{j=1}^d f_j(x_j) + \sum_{1 \leq k < l \leq d} f_{k,l}(x_k, x_l) + \cdots + f_{1,\dots,d}(x_1, \dots, x_d). \quad (2.2)$$

Si on suppose de plus que f_0 est constante et que les intégrales de chaque terme de la décomposition par rapport à leurs variables respectives vérifient :

$$\int_{[0,1]} f_{j_1,\dots,j_s}(x_{j_1}, \dots, x_{j_s}) dx_{j_k} = 0, \quad 1 \leq k \leq s, \quad (j_1, \dots, j_s) \in \mathcal{D}^s, \quad (2.3)$$

alors la décomposition (2.2) est unique. Une conséquence de (2.2) et (2.3) est que les termes de la décomposition sont deux à deux orthogonaux. Autrement dit, pour $(k_1, \dots, k_s) \neq (l_1, \dots, l_s)$, $(k_1, \dots, k_s) \in \mathcal{D}^s$, $(l_1, \dots, l_s) \in \mathcal{D}^s$, on a :

$$\int_{[0,1]^d} f_{k_1,\dots,k_s}(x_{k_1}, \dots, x_{k_s}) f_{l_1,\dots,l_s}(x_{l_1}, \dots, x_{l_s}) d\mathbf{x} = 0,$$

car au moins un des indices $k_1, \dots, k_s$, $l_1, \dots, l_s$, n'est pas répété et l'intégrale associée à cet indice vaut 0 d'après (2.3).

Une deuxième conséquence est que f_0 est donné par :

$$f_0 = \int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x},$$

qui n'est autre que la moyenne μ de f .

Dans le contexte de l'analyse de sensibilité, la décomposition ANOVA (2.2) sert à quantifier l'importance de chacun des termes issus de la décomposition sur la sortie Y du modèle. La quantification de cette importance utilise la construction d'indicateurs statistiques appelés indices de Sobol'.

2.1.3 Indices de Sobol'

Sobol' montre que tous les termes de la décomposition ANOVA (2.2) peuvent être évalués par :

$$\begin{aligned} f_j(x_j) &= -\mu + \int_{[0,1]^{d-1}} f(\mathbf{x}) d\mathbf{x}_{-j}, \\ f_{k,l}(x_k, x_l) &= -\mu - f_k(x_k) - f_l(x_l) + \int_{[0,1]^{d-2}} f(\mathbf{x}) d\mathbf{x}_{-\{k,l\}}, \\ &\vdots \end{aligned}$$

L'ensemble de ces égalités peut se réécrire sous la forme :

$$f_u(\mathbf{X}_u) = E[Y|\mathbf{X}_u] - \sum_{v \subset u} f_v(\mathbf{X}_v), \quad u \subset \mathcal{D}.$$

En élevant au carré puis en intégrant (2.2), par orthogonalité des termes, on obtient la décomposition de la variance $\sigma^2 = \text{Var}[Y]$ suivante :

$$\sigma^2 = \sum_{j=1}^d \sigma_j^2 + \sum_{1 \leq k < l \leq d} \sigma_{k,l}^2 + \cdots + \sigma_{1,\dots,d}^2, \quad (2.4)$$

où

$$\sigma_u^2 = \text{Var}[E[Y|\mathbf{X}_u]] - \sum_{v \subset u} \sigma_v^2, \quad u \subset \mathcal{D}.$$

La définition des indices de Sobol' est issue de la décomposition de la variance (2.4).

On définit :

i) l'indice de Sobol' élémentaire :

$$S_u = \frac{\sigma_u^2}{\sigma^2}, \quad u \subset \mathcal{D}.$$

Si $u = \{j\}$, S_j est appelé indice d'ordre un et quantifie l'effet principal dû à la variable X_j . Si $\#u > 1$, S_u quantifie l'effet de l'interaction d'ordre $\#u$ entre les variables X_j , $j \in u$.

ii) l'indice de Sobol' fermé d'ordre $\#u$:

$$\underline{S}_u = \frac{\tau_u^2}{\sigma^2}, \quad u \subset \mathcal{D}, \quad (2.5)$$

où :

$$\tau_u^2 = \sum_{v \subseteq u} \sigma_v^2.$$

$\underline{S}_u$ quantifie l'effet principal du groupe de variables $\mathbf{X}_u$. Cet effet correspond à la somme de tous les effets principaux des variables X_j , $j \in u$ et de toutes les interactions entre les variables X_j .

iii) l'indice de Sobol' total :

$$\overline{S}_u = \frac{\bar{\tau}_u^2}{\sigma^2}, \quad u \subset \mathcal{D}, \quad (2.6)$$

où :

$$\bar{\tau}_u^2 = \sum_{v \cap u \neq \emptyset} \sigma_v^2.$$

$\overline{S}_u$ quantifie l'effet principal du groupe de variables $\mathbf{X}_u$ ainsi que de toutes les interactions faisant intervenir une ou plusieurs variables X_j , $j \in u$.

Tous ces indices sont des scalaires compris entre 0 et 1 et s'interprètent de la façon suivante : plus l'indice est proche de 1, plus la variable (ou le groupe de variables) associée à cet indice est influente sur la sortie Y . À l'inverse, plus l'indice est proche de 0, moins cette variable est influente. Pour $u \subset \mathcal{D}$, nous avons les relations suivantes :

$$\bar{\tau}_u^2 = \sigma^2 - \underline{\tau}_{-u}^2, \quad 0 \leq \sigma_u^2 \leq \underline{\tau}_u^2 \leq \bar{\tau}_u^2 \leq \sigma^2.$$

À titre d'exemple, considérons un modèle f à trois variables X_1, X_2, X_3 . Alors :

$$\begin{aligned} \underline{S}_{\{1,2\}} &= S_1 + S_2 + S_{\{1,2\}}, \\ \bar{S}_1 &= S_1 + S_{\{1,2\}} + S_{\{1,3\}} + S_{\{1,2,3\}}. \end{aligned}$$

Pour le cas particulier $\#u = 1$, l'indice de Sobol' fermé s'identifie à l'indice élémentaire et on parle alors d'indices de Sobol' d'ordre un.

2.2 Introduction aux méthodes d'estimation

Lorsqu'une expression analytique du modèle est à disposition, le calcul des indices de Sobol' devient un problème de calcul d'intégrales qui peut être résolu à l'aide d'un logiciel de calcul formel (Maple ou Mathematica par exemple). Nous avons développé un code Maple permettant de calculer les indices d'ordre un, les indices fermés d'ordre deux et les indices totaux d'une fonction pour laquelle on dispose d'une expression analytique et pour des lois de distribution uniformes sur $[0, 1]$ des variables d'entrée. Le code est présenté dans l'annexe C.

Cependant, bien souvent les modèles mettent en jeu des systèmes d'équations complexes, auxquels viennent s'ajouter des lois de distributions des paramètres potentiellement non uniformes, formant un tout s'avérant impossible à formaliser analytiquement. Dans ce cas, il devient nécessaire de faire appel à des méthodes d'estimation des indices de Sobol'.

Dans cette section nous présentons quelques méthodes Monte Carlo permettant d'estimer ces indices en commençant par la méthode originelle introduite par Sobol' dans [Sob67].

2.2.1 Méthode de Sobol'

Nous nous focalisons ici sur l'estimation des indices de Sobol' fermés et totaux : $\underline{S}_u$ et $\bar{S}_u$, $u \subset \mathcal{D}$. La méthode d'estimation proposée par Sobol' consiste en une méthode Monte Carlo dite *pick and freeze*. Dans l'approche *pick and freeze*, l'indice de Sobol' est perçu comme le coefficient de régression entre la sortie du modèle et sa réplique *pick-freezed*. Pour l'estimation de $\underline{S}_u$ (respectivement $\bar{S}_u$), cette réplique est obtenue en gelant la valeur de la variable $\mathbf{X}_u$ (respectivement $\mathbf{X}_{-u}$), puis en rééchantillonnant les autres variables $\mathbf{X}_{-u}$ (respectivement $\mathbf{X}_u$).

L'estimation des indices fermés et totaux nécessite à la fois d'évaluer les numérateurs respectifs $\underline{\tau}_u^2$ et $\bar{\tau}_u^2$ et la variance σ^2 . L'estimation de σ^2 est simplement obtenue en considérant l'estimateur empirique :

$$\widehat{\sigma^2} = \frac{1}{n} \sum_{i=1}^n (f(\mathbf{X}_i) - \hat{\mu})^2, \quad \hat{\mu} = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i).$$

Notons $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{Z}_i\}_{i=1}^n$. L'estimation de $\underline{\tau}_u^2$ requiert la construction d'un plan d'expériences $\mathcal{P}_u = \{\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}\}$. En pratique, $\mathcal{P}_u$ est obtenu en substituant les colonnes indexées par $-u$ de $\mathcal{P}$ par les colonnes correspondantes de $\mathcal{P}'$. Le premier estimateur de $\underline{\tau}_u^2$ a été introduit par Sobol' dans [Sob93] et est défini par :

$$\widehat{\underline{\tau}}_u^2 = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i) f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - \hat{\mu}^2, \quad u \subset \mathcal{D}. \quad (2.7)$$

L'estimation de $\bar{\tau}_u^2$ requiert la construction d'un plan d'expériences $\mathcal{P}_{-u} = \{\mathbf{X}_{i,-u} : \mathbf{Z}_{i,u}\}$. En pratique, $\mathcal{P}_{-u}$ est obtenu en substituant les colonnes indexées par u de $\mathcal{P}$ par les colonnes correspondantes de $\mathcal{P}'$. L'estimateur de $\bar{\tau}_u^2$ a été introduit par Homma et Saltelli dans [HS96] et est défini par :

$$\widehat{\bar{\tau}}_u^2 = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i)^2 - \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i) f(\mathbf{X}_{i,-u} : \mathbf{Z}_{i,u}), \quad u \subset \mathcal{D}. \quad (2.8)$$

Pour $u \subset \mathcal{D}$, en observant les formules (2.7) et (2.8), on remarque que l'estimation d'un indice fermé $\underline{S}_u$ ou d'un indice total $\bar{S}_u$ nécessite d'évaluer le modèle f aux points hybrides $\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}$ ou $\mathbf{X}_{i,-u} : \mathbf{Z}_{i,u}$.

En s'appuyant sur les formules précédentes, on constate que :

- i) pour $\#u = \{j\}$, $j \in \mathcal{D}$, l'estimation de tout les indices d'ordre un $\underline{S}_j$ ou de tout les indices totaux $\bar{S}_j$ requiert $n(d+1)$ évaluations du modèle f ,
- ii) pour $\#u > 1$, l'estimation de tous les indices fermés d'ordre $\#u$ $\underline{S}_u$ requiert $n \left(\binom{d}{\#u} + 1 \right)$ évaluations du modèle f .

Pour synthétiser, la méthode de Sobol' possède les caractéristiques suivantes :

- les hypothèses d'utilisation de la méthode sont peu contraignantes, il est seulement demandé que la fonction f soit de carré intégrable par rapport à la loi produit des X_j , $j \in \mathcal{D}$; c'est à dire $E[Y^2] < +\infty$.
- la vitesse de convergence de la méthode est indépendante de la dimension d , toutefois elle est seulement d'ordre $\mathcal{O}(n^{-1/2})$ (vitesse du TLC) pour une taille n des plans d'expériences construits.

Des améliorations de ces stratégies d'estimation sont possibles afin de réduire le nombre total d'évaluations du modèle. Nous décrivons ces améliorations dans la section suivante.

2.2.2 Extensions de Saltelli

Nous avons évoqué dans la section précédente deux stratégies d'estimation permettant d'estimer tous les indices fermés $\underline{S}_u$ d'ordre $\#u$ ou tous les indices totaux $\overline{S}_u$ pour $\#u = 1$. Dans [Sal02], Saltelli propose deux nouvelles stratégies d'estimation étendant les résultats de Sobol'. Puis dans [Owe12], Owen généralise les résultats de Saltelli à l'aide d'un formalisme combinatoire portant sur les sous-ensemble $u \subset \mathcal{D}$.

Nous proposons de décrire brièvement ce formalisme à l'aide des notations introduites par Owen. Nous détaillons ensuite les stratégies d'estimation proposées par Saltelli.

La différence symétrique de deux ensembles $(v, w) \subset \mathcal{D}^2$ est définie par :

$$v \Delta w = v \cup w - v \cap w.$$

L'idée sous-jacente au formalisme introduit par Owen consiste à généraliser les expressions de $\underline{\tau}_u^2$ et $\overline{\tau}_u^2$ à l'aide de la formule suivante :

$$\mathbb{E}[f(\mathbf{X}_v : \mathbf{Z}_{-v})f(\mathbf{X}_w : \mathbf{Z}_{-w})] = \mu^2 + \underline{\tau}_{-(v \Delta w)}^2, \quad (v, w) \subset \mathcal{D}^2.$$

dont un estimateur naturel est obtenu par :

$$\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i,v} : \mathbf{Z}_{i,-v})f(\mathbf{X}_{i,w} : \mathbf{Z}_{i,-w}) = \widehat{\mu}^2 + \widehat{\underline{\tau}}_{-(v \Delta w)}^2, \quad (v, w) \subset \mathcal{D}^2, \quad (2.9)$$

Remarquons maintenant qu'en posant $v = \mathcal{D}$ et $w = u$ on obtient :

$$\begin{aligned} -(v \Delta w) &= u, \\ \mathbf{X}_{i,v} : \mathbf{Z}_{i,-v} &= \mathbf{X}_{i,\mathcal{D}} : \mathbf{Z}_{i,\emptyset} = \mathbf{X}_i, \\ \mathbf{X}_{i,w} : \mathbf{Z}_{i,-w} &= \mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}. \end{aligned}$$

On retrouve alors l'expression (2.7) :

$$\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i)f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) = \widehat{\mu}^2 + \widehat{\underline{\tau}}_u^2, \quad u \subset \mathcal{D}.$$

Par un même raisonnement, en posant cette fois $v = \mathcal{D}$ et $w = -u$ on obtient :

$$\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i)f(\mathbf{X}_{i,-u} : \mathbf{Z}_{i,u}) = \widehat{\mu}^2 + \widehat{\underline{\tau}}_{-u}^2, \quad u \subset \mathcal{D},$$

or comme $\overline{\tau}_u^2 = \sigma^2 - \underline{\tau}_{-u}^2$ on retrouve alors l'expression (2.8). Ces deux cas sont des cas particuliers de l'indice de Sobol' généralisé (ISG) introduit par Owen. Nous ne donnons pas plus de précision concernant cet indice dont l'analyse est développée dans [Owe12].

A partir du formalisme combinatoire d'Owen, on peut extraire les stratégies d'estimation de Saltelli. Pour cela, considérons la table 2.1 décrivant les ensembles $-(v \Delta w)$ pour $(v, w) \in \{\emptyset, k, l, -k, -l, \mathcal{D}\}^2$. La lecture de cette table se fait avec l'équation (2.9) :

TABLE 2.1 – Ensembles $-(v\Delta w)$ pour $(v, w) \in \{\emptyset, k, l, -k, -l, \mathcal{D}\}^2$.

$v \setminus w$	$\emptyset$	k	l	$-k$	$-l$	$\mathcal{D}$
$\emptyset$	$\mathcal{D}$					
k	$-k$	$\mathcal{D}$				
l	$-l$	$-\{k, l\}$	$\mathcal{D}$			
$-k$	k	$\emptyset$	$\{k, l\}$	$\mathcal{D}$		
$-l$	l	$\{k, l\}$	$\emptyset$	$-\{k, l\}$	$\mathcal{D}$	
$\mathcal{D}$	$\emptyset$	k	l	$-k$	$-l$	$\mathcal{D}$

- $-(v\Delta w) = \emptyset$ signifie que le couple (v, w) permet d'estimer μ^2 ($\tau_{\emptyset}^2 = 0$).
- $-(v\Delta w) = \mathcal{D}$ signifie que le couple (v, w) permet d'estimer $\tau_{\mathcal{D}}^2 = \sigma^2$.
- $-(v\Delta w) = k$ (ou l ou $\{k, l\}$) signifie que le couple (v, w) permet d'estimer τ_k^2 (ou τ_l^2 ou $\tau_{\{k, l\}}^2$).
- $-(v\Delta w) = -k$ (ou $-l$ ou $-\{k, l\}$) signifie que le couple (v, w) permet d'estimer $\bar{\tau}_k^2$ (ou $\bar{\tau}_l^2$ ou $\bar{\tau}_{\{k, l\}}^2$) étant donné σ^2 .

La première stratégie de Saltelli correspond au théorème 1 dans [Sal02] :

Théorème 2.1. *Etant donné $(v, w) \in \{\emptyset, 1, 2, \dots, d, \mathcal{D}\}^2$, il est possible de calculer à l'aide de (2.9) une estimation de :*

- chaque indice de Sobol' d'ordre un $\underline{S}_j$, $j \in \mathcal{D}$,
- chaque indice de Sobol' total $\bar{S}_j$, $j \in \mathcal{D}$.

Le nombre total d'évaluations du modèle vaut $n(d + 2)$.

La deuxième stratégie de Saltelli correspond au théorème 2 dans [Sal02] :

Théorème 2.2. *Etant donné $(v, w) \in \{\emptyset, 1, 2, \dots, d, -1, -2, \dots, -d, \mathcal{D}\}^2$, il est possible de calculer à l'aide de (2.9) deux estimations de :*

- chaque indice $\underline{S}_j$ et $\bar{S}_j$, $j \in \mathcal{D}$,
- chaque indice $\underline{S}_{\{k, l\}}$ et $\bar{S}_{\{k, l\}}$, $(k, l) \in \mathcal{D}^2$,

Le nombre total d'évaluations du modèle vaut $n(2d + 2)$.

Les deux théorèmes proposés par Saltelli offrent la possibilité d'estimer plusieurs groupes d'indices pour un coût total relativement faible. En parallèle des extensions de Saltelli et Owen, des formules alternatives pour les estimateurs (2.7) et (2.8) ont été proposées afin d'améliorer la précision de ces estimateurs. La section qui suit est dédiée à l'étude de ces formules alternatives.

2.2.3 Autres estimateurs

Les premières formules alternatives concernent l'estimateur $\widehat{\tau}_u^2$. Dans [Sob93], Sobol' remarque qu'il est difficile d'estimer des petites valeurs de τ_u^2 via (2.7). En pratique, on observe un biais lorsque τ_u^2 prend des petites valeurs. Les formules qui suivent ont été proposées dans l'optique de pallier ce problème de biais observé. Dans [Owe13a], Owen propose des résultats asymptotiques ainsi qu'une étude empirique approfondie de ces différents estimateurs.

La première formule alternative est proposée par Saltelli [Sal02]. Cette formule consiste à remplacer le terme $f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})$ par $f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - f(\mathbf{Z}_i)$ faisant ainsi disparaître le terme $\hat{\mu}$. Cet estimateur, appelé *Correlation 1* par Owen, prend la forme suivante :

$$\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i) (f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - f(\mathbf{Z}_i)), \quad u \subset \mathcal{D}.$$

Dans [SM08], Sobol' et Myshetskaya montrent que même l'estimateur précédent continue d'avoir une variance élevée quand μ est grand. Ces deux auteurs proposent alors une seconde alternative consistant à remplacer le terme $f(\mathbf{X}_i)$ par $f(\mathbf{X}_i) - \hat{\mu}$. Cet estimateur est appelée *Oracle 1* par Owen et se définit comme suit :

$$\frac{1}{n} \sum_{i=1}^n (f(\mathbf{X}_i) - \hat{\mu}) (f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - f(\mathbf{Z}_i)), \quad u \subset \mathcal{D}.$$

Owen introduit également deux nouveaux estimateurs appelés respectivement *Oracle 2* et *Correlation 2* :

$$\frac{1}{n} \sum_{i=1}^n (f(\mathbf{X}_i) - \hat{\mu}) (f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - \hat{\mu}), \quad u \subset \mathcal{D},$$

$$\frac{1}{n} \sum_{i=1}^n (f(\mathbf{X}_i) - f(\mathbf{W}_{i,u} : \mathbf{X}_{i,-u})) (f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - f(\mathbf{Z}_i)), \quad u \subset \mathcal{D},$$

où $\mathbf{W}$ est un vecteur identiquement distribué suivant la loi de $\mathbf{X}$ et indépendant de $\mathbf{X}$. L'estimateur *Oracle 2* introduit le terme $\hat{\mu}$ dans chacun des deux membres. L'estimateur *Correlation 2* requiert de construire un plan d'expériences supplémentaire pour l'estimation de τ_u^2 .

L'étude empirique menée par Owen, révèle que l'estimateur *Oracle 2* est le plus performant lorsque la valeur de τ_u^2 est élevée. Il apparait également que la précision de *Correlation 2* est supérieure à tous les autres estimateurs lorsque τ_u^2 est proche de zéro (< 0.1).

Dans [JKL⁺14], Janon *et al.* étudient deux estimateurs de l'indice $\underline{S}_u$, notés $\widetilde{\underline{S}}_u$ et $\widehat{\underline{T}}_u$. Chacun s'appuie sur une réécriture de $\underline{S}_u$ de la manière suivante :

$$\underline{S}_u = \frac{\text{Cov}[f(\mathbf{X})f(\mathbf{X}_u : \mathbf{Z}_{-u})]}{\text{Var}[f(\mathbf{X})]}.$$

Cette réécriture provient de l'égalité $\text{Var}[\mathbb{E}[f(\mathbf{X})|\mathbf{X}_u]] = \text{Cov}[f(\mathbf{X})f(\mathbf{X}_u : \mathbf{Z}_{-u})]$ due au fait que $f(\mathbf{X})$ et $f(\mathbf{X}_u : \mathbf{Z}_{-u})$ sont identiquement distribuées et indépendantes conditionnellement à $\mathbf{X}_u$. Cette réécriture se perçoit comme un coefficient de régression entre la sortie du modèle $f(\mathbf{X})$ et sa réplique *pick-freezed* $f(\mathbf{X}_u : \mathbf{Z}_{-u})$. En la développant on obtient :

$$\underline{S}_u = \frac{\mathbb{E}[f(\mathbf{X})f(\mathbf{X}_u : \mathbf{Z}_{-u})] - \mathbb{E}[f(\mathbf{X})]\mathbb{E}[f(\mathbf{X}_u : \mathbf{Z}_{-u})]}{\mathbb{E}[f(\mathbf{X})^2] - \mathbb{E}[f(\mathbf{X})]^2}. \quad (2.10)$$

Ainsi, il est possible de réécrire l'indice de Sobol' $\underline{S}_u$ comme une fonction de plusieurs moyennes. Le premier estimateur $\widetilde{\underline{S}}_u$ est défini par :

$$\widetilde{\underline{S}}_u = \frac{\widetilde{\tau}_u^2}{\widetilde{\sigma}^2}, \quad u \subset \mathcal{D}, \quad (2.11)$$

$$\text{où : } \begin{cases} \widetilde{\tau}_u^2 = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i)f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - \left(\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i) \right) \left(\frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) \right), \\ \widetilde{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n \left(f(\mathbf{X}_i) - \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i) \right)^2. \end{cases}$$

Le second estimateur $\widehat{\underline{T}}_u$, introduit dans [MN^M06], est défini par :

$$\widehat{\underline{T}}_u = \frac{\widehat{\tau}_u^2}{\widehat{\sigma}^2}, \quad u \subset \mathcal{D}, \quad (2.12)$$

$$\text{où : } \begin{cases} \widehat{\tau}_u^2 = \frac{1}{n} \sum_{i=1}^n f(\mathbf{X}_i)f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u}) - \left(\frac{1}{n} \sum_{i=1}^n \frac{f(\mathbf{X}_i) + f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})}{2} \right)^2, \\ \widehat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^n \frac{f(\mathbf{X}_i)^2 + f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})^2}{2} - \left(\frac{1}{n} \sum_{i=1}^n \frac{f(\mathbf{X}_i) + f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})}{2} \right)^2. \end{cases}$$

Dans [JKL⁺14], les auteurs montrent que les estimateurs (2.11) et (2.12) sont fortement consistants et asymptotiquement normaux. Le théorème suivant synthétise ces résultats :

Théorème 2.3. Notons $Y = f(\mathbf{X})$ et $Y_u = f(\mathbf{X}_u : \mathbf{Z}_{-u})$. Si $\mathbb{E}[Y^4] < +\infty$, alors pour tout $u \subset \mathcal{D}$ on a :

$$\begin{aligned} \sqrt{n}(\widetilde{\underline{S}}_u - \underline{S}_u) &\xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_1^2), \\ \sqrt{n}(\widehat{\underline{T}}_u - \underline{S}_u) &\xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_2^2). \end{aligned}$$

avec

$$\sigma_1^2 = \frac{\text{Var}[(Y - \mathbb{E}[Y])(Y_u - \mathbb{E}[Y]) - \underline{S}_u(Y - \mathbb{E}[Y])^2]}{\text{Var}[Y]^2},$$

et

$$\sigma_2^2 = \frac{\text{Var}[(Y - \mathbb{E}[Y])(Y_u - \mathbb{E}[Y]) - \underline{S}_u/2((Y - \mathbb{E}[Y])^2 + (Y_u - \mathbb{E}[Y])^2)]}{\text{Var}[Y]^2},$$

Démonstration. Nous renvoyons à la preuve de la Proposition 2.2 dans [JKL⁺14]. $\square$

De plus, les auteurs montrent que l'estimateur $\widehat{T}_u$ est asymptotiquement efficace parmi une classe d'estimateurs réguliers construite sur deux matrices d'échantillonnage.

2.3 Introduction à la méthode répliquée

Les stratégies d'estimation des indices de Sobol' proposées par Sobol' ou Saltelli possèdent toutes un même inconvénient. Le coût de ces stratégies dépend de la dimension d de l'espace des variables d'entrée. Pour de grandes valeurs de d , ces stratégies deviennent très coûteuses en terme de nombre d'évaluations du modèle.

Une alternative à ces approches consiste à construire uniquement deux plans d'expériences et à manipuler leur structure interne afin d'estimer des ensembles d'indices pour un coût indépendant de la dimension d . Cette alternative est appelée méthode à base de plans répliqués que l'on abrège par méthode répliquée. Nous commençons par définir le concept de plans répliqués.

2.3.1 Plans d'expériences répliqués

Le concept de plans d'expériences répliqués a été introduit par McKay avec sa définition d'hypercube latin répliqué [McK95]. Toutefois, ce concept peut être généralisé à d'autres structures. Pour ce faire nous introduisons la définition générale suivante :

Définition 2.2. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ deux plans d'expériences où $\mathbf{X}_i$, $\mathbf{X}'_i$ sont indépendants et identiquement distribués. $\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués si $\forall j \in \mathcal{D}$ il existe une permutation π_j de $\{1, \dots, n\}$ tel que :

$$\mathbf{X}'_{\pi_j(i),j} = \mathbf{X}_{i,j}, \quad \forall i \in \{1, \dots, n\}.$$

Si $\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués, alors pour chaque $j \in \mathcal{D}$ fixé, les colonnes j de $\mathcal{P}$ et $\mathcal{P}'$ contiennent chacune le même ensemble de valeurs. Considérons les deux plans suivants :

$$\mathcal{P} = \begin{pmatrix} 0.08 & 0.46 & 0.21 \\ 0.15 & 0.77 & 0.43 \\ 0.89 & 0.30 & 0.05 \\ 0.70 & 0.23 & 0.95 \end{pmatrix}, \quad \mathcal{P}' = \begin{pmatrix} 0.89 & 0.30 & 0.95 \\ 0.15 & 0.23 & 0.21 \\ 0.70 & 0.46 & 0.43 \\ 0.08 & 0.77 & 0.05 \end{pmatrix}.$$

$\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués et la permutation $\pi_1 = (4, 2, 1, 3)$ réordonne les éléments de la première colonne de $\mathcal{P}'$ dans l'ordre des éléments de la première colonne de $\mathcal{P}$.

Les hypercubes latins répliqués de McKay s'identifient à des plans répliqués. L'utilisation de r hypercubes latins répliqués est la caractéristique principale de la méthode d'estimation proposée par McKay dans [McK95]. Avec cette méthode, le coût total pour estimer tous les indices de Sobol' d'ordre un est de rn . Ce coût est très intéressant car il est indépendant de la dimension d de l'espace des variables d'entrée. Toutefois, afin de débiaiser l'estimateur proposé dans sa méthode, McKay suggère de choisir une valeur de r relativement élevée. Une description détaillée de la méthode de McKay est proposée dans [Tis12].

La méthode répliquée s'appuie sur la même idée que la méthode de McKay qui est d'obtenir un coût total d'estimation indépendant de la dimension d . Cependant, la méthode répliquée se distingue par l'utilisation d'un estimateur différent requérant seulement deux plans répliqués et faisant ainsi disparaître la contrainte du choix de la valeur de r .

2.3.2 Méthode répliquée

La méthode répliquée consiste à combiner l'utilisation de deux plans répliqués avec les estimateurs *pick and freeze* de Sobol'. Cette méthode est d'abord introduite par Mara et Rakoto Joseph dans [MJ08] pour l'estimation de tous les indices de Sobol' d'ordre un, puis est étendue dans [TP15] par l'utilisation d'hypercubes latins répliqués. Nous commençons par rappeler la définition d'hypercubes latins répliqués telle qu'introduite dans [TP15] :

Définition 2.3. Soit Π_n l'ensemble des permutations de $\{1, \dots, n\}$. On dit que $\{\mathbf{X}_i\}_{i=1}^n$ et $\{\mathbf{X}'_i\}_{i=1}^n$ sont deux hypercubes latins répliqués si :

$$\begin{aligned} \mathbf{X}_i &= \left(\frac{\pi_1^{(1)}(i) - U_{\pi_1^{(1)}(i),1}}{n}, \dots, \frac{\pi_d^{(1)}(i) - U_{\pi_d^{(1)}(i),d}}{n} \right), \\ \mathbf{X}'_i &= \left(\frac{\pi_1^{(2)}(i) - U_{\pi_1^{(2)}(i),1}}{n}, \dots, \frac{\pi_d^{(2)}(i) - U_{\pi_d^{(2)}(i),d}}{n} \right), \end{aligned} \quad (2.13)$$

où les $\pi_j^{(1)}$, les $\pi_j^{(2)}$ et les $U_{i,j}$ sont des variables aléatoires indépendantes et uniformément distribuées respectivement sur Π_n , Π_n et $[0, 1]$.

Nous détaillons ci-dessous la méthode d'estimation de tous les indices d'ordre un S_j , $j \in \mathcal{D}$, basée sur l'utilisation de deux hypercubes latins répliqués. Nous illustrons cette méthode à l'aide d'un exemple.

Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ deux hypercubes latins répliqués construits suivant (2.13). Dans notre exemple, on fixe $n = 4$ et $d = 3$, les deux hypercubes

latins répliqués sont :

$$\mathcal{P} = \begin{pmatrix} 0.35 & 0.45 & 0.78 \\ 0.09 & 0.20 & 0.41 \\ 0.91 & 0.72 & 0.24 \\ 0.63 & 0.90 & 0.68 \end{pmatrix}, \quad \mathcal{P}' = \begin{pmatrix} 0.35 & 0.72 & 0.41 \\ 0.91 & 0.90 & 0.68 \\ 0.63 & 0.20 & 0.78 \\ 0.09 & 0.45 & 0.24 \end{pmatrix}.$$

On note par $\{f(\mathbf{X}_i)\}_{i=1}^n$ et $\{f(\mathbf{X}'_i)\}_{i=1}^n$ l'ensemble des évaluations obtenues à partir de ces deux plans. Comme $\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués, d'après la définition 2.2, $\forall j \in \mathcal{D}$, il existe une permutation π_j permettant de réordonner les éléments de la colonne j de $\mathcal{P}'$ dans l'ordre des éléments de la colonne j de $\mathcal{P}$.

Dans notre exemple les permutations π_j sont les suivantes :

$$\pi_1 = \begin{pmatrix} 1 \\ 4 \\ 2 \\ 3 \end{pmatrix}, \quad \pi_2 = \begin{pmatrix} 4 \\ 3 \\ 1 \\ 2 \end{pmatrix}, \quad \pi_3 = \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix}.$$

Le point clé de la méthode répliquée consiste à utiliser les permutations π_j afin d'imiter l'évaluation du modèle f au point hybride $X_{i,j} : \mathbf{Z}_{i,-j}$ dans les formules (2.7), (2.11) ou (2.12). Cela est obtenu en réordonnant l'ensemble d'évaluations $\{f(\mathbf{X}'_i)\}_{i=1}^n$ par la permutation π_j . Soit $j \in \mathcal{D}$, $\forall i \in \{1, \dots, n\}$ on a :

$$\pi_j(f(\mathbf{X}'_i)) = f(\mathbf{X}'_{\pi_j(i)}) = f(X'_{\pi_j(i),j} : \mathbf{X}'_{\pi_j(i),-j}),$$

puis par la définition 2.2 :

$$f(X'_{\pi_j(i),j} : \mathbf{X}'_{\pi_j(i),-j}) = f(X_{i,j} : \mathbf{X}'_{\pi_j(i),-j}),$$

ainsi :

$$\pi_j(f(\mathbf{X}'_i)) = f(X_{i,j} : \mathbf{X}'_{\pi_j(i),-j}).$$

Réordonner l'ensemble d'évaluations $\{f(\mathbf{X}'_i)\}_{i=1}^n$ par la permutation π_j revient à réordonner les lignes de $\mathcal{P}'$ par π_j et à considérer les évaluations correspondantes. Afin d'illustrer ce procédé, reprenons notre exemple et réarrangeons les lignes de $\mathcal{P}'$ par la permutation π_1 . Notons par $\mathcal{P}'_{\pi_1}$ le plan résultant. On obtient :

$$\mathcal{P} = \begin{pmatrix} 0.35 & 0.45 & 0.78 \\ 0.09 & 0.20 & 0.41 \\ 0.91 & 0.72 & 0.24 \\ 0.63 & 0.90 & 0.68 \end{pmatrix}, \quad \mathcal{P}'_{\pi_1} = \begin{pmatrix} 0.35 & 0.72 & 0.41 \\ 0.09 & 0.45 & 0.24 \\ 0.91 & 0.90 & 0.68 \\ 0.63 & 0.20 & 0.78 \end{pmatrix}.$$

En regardant les deux plans on observe que la première colonne est identique, c'est à dire que la variable X_1 est gelée. On observe également que pour chaque ligne i de

$\mathcal{P}'_{\pi_1}$, les autres variables $\mathbf{X}'_{\pi_j(i),-j}$ (couleur grise) prennent des valeurs différentes, comme-ci elles avaient été rééchantillonnées.

Pour $u = \{j\}$, l'indice de Sobol' d'ordre un S_j est alors estimé par une des formules (2.7), (2.11) ou (2.12) en substituant chaque terme $f(X_{i,j} : \mathbf{Z}_{i,-j})$ par $\pi_j(f(\mathbf{X}'_i))$.

En répétant ce même procédé pour chaque permutation π_j , $j \in \mathcal{D}$, la méthode répliquée permet d'estimer tous les indices d'ordre un pour un coût égal à $2n$ ne dépendant que de la taille n des deux plans répliqués $\mathcal{P}$ et $\mathcal{P}'$.

Propriétés asymptotiques

Dans [TP15], Tissot *et al.* étudient les propriétés asymptotiques de la méthode répliquée pour l'estimation des indices fermés $\underline{S}_u$, $u \subset \mathcal{D}$, en combinant l'utilisation de deux hypercubes latins répliqués et les estimateurs (2.11) et (2.12).

Soient $u \subset \mathcal{D}$, $\{\mathbf{X}_i\}_{i=1}^n$ un hypercube latin construit suivant (1.3) et $\{\mathbf{W}_i\}_{i=1}^n$, $\{\mathbf{W}'_i\}_{i=1}^n$ deux hypercubes latins répliqués construits suivant (2.13). Soient les deux ensembles d'évaluations :

$$\{f(\mathbf{X}_{i,u} : \mathbf{W}_{i,-u})\}_{i=1}^n, \quad \{f(\mathbf{X}_{i,u} : \mathbf{W}'_{i,-u})\}_{i=1}^n.$$

On note par $\widetilde{\underline{S}}_j^{RLHS} = \widetilde{\tau}_u^{RLHS} / \sigma^{RLHS}$ et $\widehat{\underline{T}}_j^{RLHS} = \widehat{\tau}_u^{RLHS} / \sigma^{RLHS}$ les estimateurs (2.11) et (2.12) obtenus en substituant $f(\mathbf{X}_i)$ et $f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})$ par $f(\mathbf{X}_{i,u} : \mathbf{W}_{i,-u})$ et $f(\mathbf{X}_{i,u} : \mathbf{W}'_{i,-u})$. Ces estimateurs ont les propriétés suivantes :

Proposition 2.1. *Soit $u \subset \mathcal{D}$:*

- i) *Si f^4 est intégrable, alors $\widetilde{\underline{S}}_u^{RLHS}$ et $\widehat{\underline{T}}_u^{RLHS}$ sont fortement consistants.*
- ii) *Si f^6 est intégrable, alors :*

$$\begin{aligned} \sqrt{n}(\widetilde{\underline{S}}_u^{RLHS} - \underline{S}_u) &\xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_1), \\ \sqrt{n}(\widehat{\underline{T}}_u^{RLHS} - \underline{S}_u) &\xrightarrow[n \rightarrow +\infty]{\mathcal{L}} \mathcal{N}(0, \sigma_2), \end{aligned}$$

avec σ_1 et σ_2 définis exactement comme dans le théorème 2.3.

iii) *De plus :*

$$\begin{aligned} \mathbb{E}[\widetilde{\tau}_u^{RLHS}] &= \tau_u^2 - \frac{1}{n}\tau_u^2 + B_{n,1} + B_{\#u,n}, \\ \mathbb{E}[\sigma^{RLHS}] &= \sigma^2 + B_{n,3}, \\ \mathbb{E}[\widehat{\tau}_u^{RLHS}] &= \tau_u^2 - \frac{1}{2n}\tau_u^2 + B_{n,1} + B_{n,2} + B_{\#u,n}, \\ \mathbb{E}[\widehat{\sigma}^{RLHS}] &= \sigma^2 - \frac{1}{2n}\tau_u^2 + B_{n,1} + B_{n,2}, \end{aligned}$$

où :

$$\begin{aligned} |B_{n,1}| &\leq \left(\frac{d+1}{n} + 2 \right) \left(\frac{d+1}{n} \right) \mathbb{E}[Y^2], \\ |B_{n,2}| &\leq \frac{\sigma^2}{2n}, \\ -\frac{1}{n-1}\sigma^2 &\leq |B_{n,3}| \leq 0, \\ |B_{n,\#u}| &\leq \left(\frac{d-\#u+1}{n} + 2 \right) \left(\frac{d-\#u+1}{n-1} \right) \mathbb{E}[Y^2]. \end{aligned}$$

Démonstration. Nous renvoyons à la preuve de la Proposition 3.5 dans [TP15]. $\square$

Extension

Toujours dans [TP15], les auteurs étendent la méthode répliquée à l'estimation de tous les indices fermés d'ordre deux en utilisant deux tableaux orthogonaux randomisés de force deux répliqués. La définition suivante de ces structures est la suivante :

Définition 2.4. Soit $A = \{\mathbf{A}_i\}_{i=1}^{q^2}$ un tableau orthogonal $OA(q^2, d, q, 2)$. Soit Π_q l'ensemble des permutations de $\{1, \dots, q\}$. On dit que $\{\mathbf{X}_i\}_{i=1}^{q^2}$ et $\{\mathbf{X}'_i\}_{i=1}^{q^2}$ sont deux tableaux orthogonaux randomisés de force deux répliqués si :

$$\begin{aligned} \mathbf{X}_i &= \left(\frac{\pi_1^{(1)}(A_{i,1}) - U_{\pi_1^{(1)}(A_{i,1}),1}}{q}, \dots, \frac{\pi_d^{(1)}(A_{i,d}) - U_{\pi_d^{(1)}(A_{i,d}),d}}{q} \right), \\ \mathbf{X}'_i &= \left(\frac{\pi_1^{(2)}(A_{i,1}) - U_{\pi_1^{(2)}(A_{i,1}),1}}{q}, \dots, \frac{\pi_d^{(2)}(A_{i,d}) - U_{\pi_d^{(2)}(A_{i,d}),d}}{q} \right), \end{aligned} \quad (2.14)$$

où les $\pi_j^{(1)}$, les $\pi_j^{(2)}$ et les $U_{i,j}$ sont des variables aléatoires indépendantes et uniformément distribuées respectivement sur Π_q , Π_q et $[0, 1]$.

On peut généraliser la définition 2.2 de plans répliqués, on parle alors de plans répliqués d'ordre $\#u$:

Définition 2.5. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ deux plans d'expériences où $\mathbf{X}_i$, $\mathbf{X}'_i$ sont indépendants et identiquement distribués. On note $\mathcal{P}_u = \{\mathbf{X}_{i,u}\}_{i=1}^n$ (respectivement $\mathcal{P}'_u = \{\mathbf{X}'_{i,u}\}_{i=1}^n$), $u \subset \mathcal{D}$, le sous-ensemble de colonnes de $\mathcal{P}$ (respectivement $\mathcal{P}'$) indexées par u .

$\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués d'ordre a , $a \in \mathcal{D}$, si $\forall u \subset \mathcal{D}$ de cardinalité $\#u = a$, $\mathcal{P}_u$ et $\mathcal{P}'_u$ consistent en le même ensemble de points dans $[0, 1]^a$. On note alors par π_u la permutation réordonnant les lignes de $\mathcal{P}'_u$ dans l'ordre des lignes de $\mathcal{P}_u$.

Pour estimer tous les indices fermés d'ordre deux, on procède de manière similaire au cas de l'estimation des indices d'ordre un. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^{q^2}$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^{q^2}$ deux tableaux orthogonaux randomisés de force deux répliqués construits suivant (2.14). On note par $\{f(\mathbf{X}_i)\}_{i=1}^{q^2}$ et $\{f(\mathbf{X}'_i)\}_{i=1}^{q^2}$ l'ensemble des évaluations obtenues à partir de ces deux plans.

Soit $(k, l) \in \mathcal{D}^2$, $k < l$, on note par $\mathcal{P}_k$ et $\mathcal{P}_l$ (respectivement $\mathcal{P}'_k$ et $\mathcal{P}'_l$) les colonnes k et l de $\mathcal{P}$ (respectivement $\mathcal{P}'$). Comme $\mathcal{P}$ et $\mathcal{P}'$ sont deux plans répliqués d'ordre deux, d'après la définition 2.5, il existe une permutation $\pi_{\{k,l\}}$ permettant de réordonner les lignes du couple de colonnes $\mathcal{P}'_{\{k,l\}}$ dans l'ordre des lignes du couple de colonnes $\mathcal{P}_{\{k,l\}}$.

Alors, l'indice de Sobol' fermé d'ordre deux $\underline{S}_{\{k,l\}}$ est estimé par (2.7), (2.11) ou (2.12) en substituant chaque terme $f(\mathbf{X}_{i,u} : \mathbf{Z}_{i,-u})$ par $\pi_{\{k,l\}}(f(\mathbf{X}'_i))$. Ces derniers ne sont autres que l'ensemble des évaluations $\{f(\mathbf{X}'_i)\}_{i=1}^{q^2}$ réordonné par la permutation $\pi_{\{k,l\}}$.

Ainsi la méthode répliquée permet d'estimer tous les indices fermés d'ordre deux pour un coût égal à $2q^2$ où q est soit un nombre premier soit une puissance d'un nombre premier.

Remarque 2.1. *L'étude des propriétés asymptotiques des estimateurs (2.11), (2.12) pour deux tableaux orthogonaux randomisés et répliqués de force deux reste une question ouverte. Dans [Tis12], Tissot suggère de partir des résultats de Loh sur les propriétés asymptotiques des tableaux orthogonaux de force deux [Loh08]. Cet axe de recherche n'a pas été considéré dans nos travaux de thèse.*

2.3.3 Synthèse

Nous avons présenté dans ce chapitre diverses stratégies d'estimation Monte Carlo d'indices de Sobol'. Parmi ces stratégies, celles proposées par Saltelli sont particulièrement intéressantes car elles estiment un grand nombre d'indices à la fois. Toutefois, les stratégies de Saltelli possèdent un coût linéaire en la dimension d d'évaluations du modèle ; coût qui devient contraignant lorsque d est grand.

Dans ce cas, une stratégie alternative consiste à utiliser la méthode répliquée pour laquelle le nombre total d'évaluations du modèle est indépendant de d pour l'estimation des indices d'ordre un. Pour l'estimation des indices fermés d'ordre deux, le coût est soumis à la contrainte de construction d'un tableau orthogonal linéaire de force deux (théorème 1.2). Toutefois, dans [TP15] les auteurs montrent que cette contrainte n'est pas trop forte pour des dimensions allant jusqu'à la centaine.

L'inconvénient majeur de cette approche est qu'elle n'est pas adaptée à l'estimation des indices totaux. En effet, pour une dimension d , estimer l'indice total de chaque variable d'entrée requerrait de construire un tableau orthogonal linéaire de force $d - 1$ pour lequel la taille est q^{d-1} (théorème 1.2). Une telle taille devient rapidement inadéquate pour des grandes valeurs de d . Pour des raisons identiques, la méthode répliquée ne permet pas d'utiliser les estimateurs *Oracle 1* et *Correlation*

2 introduits dans la section 2.2.3 et faisant intervenir des termes similaires à celui apparaissant dans l'estimateur de l'indice total (2.8).

Dans le cadre de notre application sur Tranus détaillée au chapitre 6, nous nous intéressons à l'estimation des effets principaux et interactions d'ordre deux des paramètres d'entrée du modèle. Ainsi, la méthode répliquée apparaît comme la plus adaptée. Les estimateurs que nous utilisons en combinaison avec cette méthode sont les estimateurs (2.11) et (2.12) respectivement introduits par Janon et Monod.

Dans le chapitre 3 nous proposons une extension de la méthode répliquée pour l'estimation des indices de Sobol' d'ordre un et fermés d'ordre deux. Cette extension nécessite d'adapter la structure des plans répliqués afin d'apporter une réponse à la questions P_1 soulevée en introduction portant sur la présence de corrélations entre les variables d'entrée.

Dans les chapitres 4 et 5, nous proposons des méthodes d'estimation récursives des indices de Sobol'. Le chapitre 4 présente une version récursive de la méthode répliquée en s'appuyant sur la construction de plans répliqués emboîtés : hypercube latin emboîté et tableau orthogonal de force deux emboîté. Dans le chapitre 5, nous allons plus loin en proposant une borne d'erreur pour l'estimation récursive des indices de Sobol' par la méthode de Saltelli du théorème 2.1 et par la méthode répliquée lorsqu'on utilise un échantillonnage de type quasi-Monte Carlo. Cette borne d'erreur sert d'indicateur pour stopper l'algorithme récursif et repose sur la propriété de clôture d'un échantillonnage quasi-Monte Carlo.

Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces

Les résultats de ce chapitre ont fait l'objet d'une publication [GAP15] dans le journal *Information and Inference : a journal of the IMA*. L'objectif principal est d'apporter une réponse à la problématique P_1 soulevée en introduction, en proposant une généralisation de la méthode répliquée pour l'estimation des indices de Sobol' par groupe introduits dans [JLD06].

Soit $X = (X_1, \dots, X_d)$, le vecteur de variables aléatoires modélisant les variables d'entrée du modèle. Lorsque les entrées sont corrélées, la décomposition ANOVA fonctionnelle présentée à la section 2.1.2 n'est plus valable dû à la non orthogonalité deux à deux des termes de la décomposition. Ainsi, la définition des indices de Sobol' qui en découle est difficilement interprétable. En particulier, il devient compliqué d'ordonner les variables d'entrée suivant leurs niveaux d'influence respectifs sur la sortie du modèle.

Nous ne proposons pas dans ce chapitre une description exhaustive des différentes approches pour réaliser une analyse de sensibilité en présence de corrélations des variables d'entrée. Nous renvoyons le lecteur intéressé aux thèses de G. Chastaing [Cha13] et Y. Caniou [Can12] pour une revue bibliographique assez complète sur le sujet.

Dans notre contexte, la corrélation entre variables d'entrée se définit par la présence de groupes de variables corrélées. Plus précisément, deux variables appartenant à un même groupe sont corrélées mais deux variables appartenant chacune à un groupe différent sont indépendantes. Le vecteur X s'écrit alors de la manière suivante :

$$X = (X_1, \dots, X_I, \underbrace{X_{I+1}, \dots, X_{I+k_1}}_{\vec{X}_{I+1}}, \dots, \underbrace{X_{I+1+\Sigma_{i-1}}, \dots, X_{I+\Sigma_i}}_{\vec{X}_{I+i}}, \\ \dots, \underbrace{X_{I+1+\Sigma_{m-1}}, \dots, X_d}_{\vec{X}_{I+m}}), \text{ où } \Sigma_i = \sum_{l=0}^i k_l, \quad k_0 = 0, \quad d = I + \Sigma_m,$$

et $X_1, \dots, X_I$ sont I variables aléatoires et indépendantes des vecteurs aléatoires indépendants $\vec{X}_{I+1}, \dots, \vec{X}_{I+m}$. Les composantes des vecteurs $\vec{X}_{I+i}$, $1 \leq i \leq m$, présentent des corrélations précisées plus loin. Comme mentionné plus haut, nous proposons dans ce chapitre une généralisation de la méthode répliquée [TP15] à l'estimation des indices groupés introduits par [JLD06] dans ce contexte d'entrées corrélées par groupe.

Nous nous intéressons plus particulièrement au cas de groupes de variables corrélées où la corrélation à l'intérieur de chaque groupe prend la forme de contraintes dites linéaires ordonnées. Ainsi, pour chaque variable $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$, une telle contrainte se traduit par :

$$X_{I+1+\Sigma_{i-1}} \leq X_{I+2+\Sigma_{i-1}} \leq \dots \leq X_{I+\Sigma_i} .$$

La loi de distribution de $\vec{X}_{I+i}$ correspond à la loi jointe des variables $X_{I+1+\Sigma_{i-1}}, \dots, X_{I+\Sigma_i}$. Pour une contrainte linéaire ordonnée, le support de cette loi jointe est une structure géométrique appelée simplexe.

Nous proposons dans cet article une méthode d'échantillonnage *space-filling* d'une loi uniforme dans un simplexe. Cette méthode repose sur une subdivision du simplexe et est appliquée pour échantillonner chaque variable multidimensionnelle $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$. Les deux plans d'expériences requis par la méthode répliquée sont des hypercubes latins et tableaux orthogonaux de force deux en dimension $I+m$. Chaque colonne de ces structures correspond soit à une variable X_i , $i \in \{1, \dots, I\}$, soit à une variable $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$. Ces dernières sont échantillonnées selon la procédure *space-filling* sur le simplexe proposée dans ce chapitre.

Nous proposons également une étude empirique des propriétés d'uniformité et de remplissage de l'espace de l'échantillonnage *space-filling*. Par ailleurs, nous comparons la performance de la méthode répliquée pour des entrées corrélées par rapport à celles de méthodes d'estimation plus classiques telles que celles de Sobol' [Sob93] et de Saltelli [Sal02].

Journal : Information and Inference : a journal of the IMA

Titre : Replication procedure for grouped Sobol' indices estimation in dependent uncertainty spaces.

Auteurs : Laurent Gilquin^{a,b}, Elise Arnaud^b, Clémentine Prieur^b.

a Inria Grenoble - Rhône-Alpes, Inovallée, 655 avenue de l'Europe, 38330 Montbonnot

*b Univ. Grenoble Alpes, Jean Kuntzmann Laboratory, F-38000 Grenoble, France
CNRS, LJK, F-38000 Grenoble, France, Inria*

3.1 Introduction

In many mathematical models, input parameters often poorly-known can have significant effects on the output of the model. It is important for model users to measure these effects. Sensitivity analysis studies how the uncertainty on an output of a mathematical model can be attributed to sources of uncertainty among the inputs. There are two main classes of sensitivity analyses called local and global sensitivity analysis. The former addresses sensitivity relatively to a nominal value of a given parameter. The latter examines sensitivity on the whole set of variation of the parameter. Global sensitivity analysis of complex and expensive mathematical models is a common practice to identify influential inputs and detect the potential interactions between them. Among the large number of available approaches, the variance-based method introduced by Sobol' [Sob93] allows to calculate sensitivity indices called Sobol' indices. Each index gives an estimation of the influence of an individual input or a group of inputs. These indices give an estimation of how the output uncertainty can be apportioned to the uncertainty in the inputs. One can distinguish first-order indices that estimate the main effect from each input or group of inputs from higher-order indices that estimate the corresponding order of interactions between inputs. Closed k -th order indices estimate up to k -th order interactions in addition to the main effect of each of the k inputs. This estimation procedure requires a significant number of model runs, number that has a polynomial growth rate with respect to the input space dimension. This cost can be prohibitive for time consuming models and only a few number of runs is not enough to retrieve accurate informations about the model inputs. Saltelli [Sal02] proposed an improvement to reduce the number of runs but the total cost still depends linearly on the dimension of input space.

The notion of replicated designs to estimate first-order Sobol' indices probably goes back to McKay [McK95] and appears later in Mara *et al.* [MJ08]. These last authors combine replicated designs with "pick-freeze" estimators [Sob93] to estimate first-order Sobol' indices. The procedure in Mara *et al.* [MJ08] has the major ad-

vantage of reducing drastically the estimation cost as the number of runs becomes independent of the input space dimension. This procedure has been further deeply studied and generalized in Tissot *et al.* [TP15] to the estimation of closed second-order indices. This generalization relies on the replication of randomized orthogonal arrays (see Lemieux [Lem09] or Owen [Owe94]).

The motivation of our paper is to extend this methodology in presence of dependent inputs. Indeed, the case of correlated parameters has to be tackled with caution, as the calculation of single input indices does not provide anymore a proper information, that can be easily interpreted. One strategy is thus to define grouped indices for groups of correlated variables as proposed in Jacques *et al.* [JLD06]. However, up to our knowledge, the problem of estimating those grouped indices, with a reasonable computing cost, has not been addressed in the literature. In this paper, we address this issue by proposing an approach based on replicated designs and randomized orthogonal arrays that enables to take into account dependency within inputs. We suppose that this dependency can be expressed through constraints. This approach can be used facing any set of constraints at the condition that one is able to provide points in the input space that verify the considered constraints. Guided by our application on a land-use and transport integrated model (LUTI) where some economical parameters are linked by order relations, this paper focus on the case of sets of linear ordered constraints. Thus we propose a sampling strategy based on the simplex geometric structure, that ensures a proper input space filling.

This paper is organized as follows. Section 3.2 recalls the notion of grouped Sobol' indices (see Jacques *et al.* [JLD06]) to study the sensitivity of an output to groups of inputs where variables within a group are dependent but variables of different groups are independent. In Section 3.3 we review the use of replication method and randomized orthogonal array to estimate first and closed second-order Sobol' indices. Section 3.4 is devoted to the sampling strategy adapted to our sets of linear ordered constraints and Section 3.5 gives a summary of the whole procedure. Numerical experimentations are conducted in Section 3.6. Classical case studies are addressed to compare our method to the ones in Sobol' [Sob93] or Saltelli [Sal02] and to assess the influence of different designs on our method.

3.2 Definition of Sobol' indices

3.2.1 Sobol' indices for independent inputs

Consider the following model defined from a black box perspective :

$$f: \begin{cases} \mathbb{R}^d & \rightarrow \mathbb{R} \\ x = (x_1, \dots, x_d) & \mapsto y = f(x) \end{cases} \quad (3.1)$$

where y is the output of the model f , x the input vector.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We model the uncertainty on the inputs by a random vector $X = (X_1, \dots, X_d)$ whose components are independent. Let

$P_X = P_{X_1} \otimes \dots \otimes P_{X_d}$ denote the distribution of X . We assume that $f \in \mathbb{L}^2(P_X)$. The model f can then be uniquely decomposed into summands of increasing dimensions (functional ANOVA decomposition [Sob93, Hoe48]) :

$$f(X) = f_0 + \sum_i f_i(X_i) + \sum_{i < j} f_{ij}(X_i, X_j) + \dots + f_{1\dots d}(X_1, \dots, X_d), \quad (3.2)$$

where each component of the decomposition verifies :

$$\int f_{i_1 \dots i_s}(x_{i_1}, \dots, x_{i_s}) dP_{X_{i_k}}(x_{i_k}) = 0, \quad \forall k \in \{1, \dots, s\}, \forall i_1, \dots, i_s \in \{1, \dots, d\}.$$

This implies that $f_0 = E[Y]$ and that the components are mutually orthogonal with respect to P_X . Let $u \subseteq \{1, \dots, d\}$, each component is defined by :

$$f_u(X_u) = E[Y|X_u] - \sum_{v \subset u} f_v(X_v).$$

The functional decomposition can be used to measure the global sensitivity of the output Y to the input X_i . By squaring and integrating (3.2), due to orthogonality we get :

$$V = \sum_i V_i + \sum_{i < j} V_{ij} + \dots + V_{1, \dots, d} \quad (3.3)$$

where

$$V_u = \text{Var}[f_u(X_u)] = \text{Var}[E[Y|X_u]] - \sum_{v \subset u} V_v$$

and

$$V = \text{Var}[Y].$$

Resulting from this decomposition, the Sobol' indices are defined by :

$$S_u = \frac{V_u}{V}.$$

Let $|u|$ denote the cardinal of u . This index measures the contribution to V of $|u|^{\text{th}}$ -order interaction between the $X_i, i \in u$.

Closed Sobol' indices are defined by :

$$\underline{S}_u = \frac{\text{Var}[E[Y|X_u]]}{V}.$$

The closed Sobol' index $\underline{S}_u$ measures the contribution of the $X_i, i \in u$, by themselves or in interaction with each other.

As an example, if there exist $i \neq j \in \{1, \dots, d\}$ such that $u = \{i, j\}$, then $\underline{S}_{ij} = S_{ij} + S_i + S_j$.

At last, note that $\sum_{u \subset \{1, \dots, d\}, u \neq \emptyset} \underline{S}_u = 1$, allowing a direct interpretation of the value of each index.

3.2.2 Sobol' indices for dependent inputs

Consider again the model (3.1). We now suppose that X is a vector of d inputs among which one or multiple groups of variables are correlated. Variables belonging to the same group are dependent but variables of different groups are independent. This implies that one variable can not appear in two groups. Each group is denoted with a multidimensional variable. X can then be defined as follows :

$$\begin{aligned} X = & (X_1, \dots, X_I, \underbrace{X_{I+1}, \dots, X_{I+k_1}}_{\vec{X}_{I+1}}, \dots, \underbrace{X_{I+1+\Sigma_{i-1}}, \dots, X_{I+\Sigma_i}}_{\vec{X}_{I+i}}, \\ & \dots, \underbrace{X_{I+1+\Sigma_{m-1}}, \dots, X_d}_{\vec{X}_{I+m}}) \text{ where } \Sigma_i = \sum_{l=0}^i k_l, \quad k_0 = 0, \end{aligned} \quad (3.4)$$

each $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$ contains k_i variables.

When inputs are dependent, the variance decomposition (3.3) does not hold as the components of the decomposition are not mutually orthogonal. The correlation of variables within $\vec{X}_{I+1}$ implies that a part of the sensitivity of one of these variables is explained by the other correlated variables. We can still compute the closed Sobol' index associated to each variable (or subset) of $\vec{X}_{I+1}$ but its value is difficult to interpret. An alternative is to define grouped Sobol' indices as in Jacques *et al.* [JLD06]. These indices are defined like in the classical case of independent inputs, but refer to the multidimensional variables. We consider X defined as in (3.4). First-order grouped Sobol' indices are defined by :

$$S_j = \frac{\text{Var}[\text{E}[Y|\vec{X}_j]]}{V}, \quad \forall j \in \{1, \dots, I+m\}.$$

If $\vec{X}_j$ is one-dimensional, then $\vec{X}_j = X_j$ and S_j has the same expression as in the independent case. If $\vec{X}_j$ is multidimensional (case $j \in \{I+1, \dots, I+m\}$), $\vec{X}_j = \vec{X}_{I+l}$, $l \in \{1, \dots, m\}$, the index is :

$$S_j = \underline{S}_{I+1+\Sigma_{l-1}, \dots, I+\Sigma_l} = \frac{\text{Var}[\text{E}[Y|X_{I+1+\Sigma_{l-1}}, \dots, X_{I+\Sigma_l}]]}{V}.$$

This index measures the impact of $\vec{X}_j$ on the output Y . Higher order Sobol' indices can also be defined. In particular, second-order grouped Sobol' indices are given by :

$$S_{j,k} = \frac{\text{Var}[\text{E}[Y|\vec{X}_j, \vec{X}_k] - \text{E}[Y|\vec{X}_j] - \text{E}[Y|\vec{X}_k]]}{V}.$$

This index measures the impact of the interaction between $\vec{X}_j$ and $\vec{X}_k$ on the output Y . The grouped Sobol' indices allow us to estimate the sensitivity of our model to the sets of linear ordered constrained inputs and the remaining unidimensional independent inputs.

3.2.3 Standard estimation of Sobol' indices

We define by design a $n \times d$ matrix which consists in n evaluations of each X_i , $i = 1, \dots, d$. The standard estimation of a closed Sobol' index $\underline{S}_u$ proposed by Sobol' requires $2n$ evaluations of the model through two designs [Sob93]. Denoting by u^c the complement of $u \subseteq \{1, \dots, d\}$, the second design of n evaluations is created from the first one by re-sampling the columns indexed by u^c . This method is referred as the Sobol' method. Let Y and Y_u be the vectors of evaluations associated to those two designs. The expression of $\underline{S}_u$ can be rewritten as follows :

$$\underline{S}_u = \frac{\mathbb{E}[YY_u] - \mathbb{E}[Y]\mathbb{E}[Y_u]}{\text{Var}[Y]} .$$

An estimator of $\underline{S}_u$ with good asymptotic properties [JKL⁺14, MNM06] is :

$$\widehat{\underline{S}}_u = \frac{\frac{1}{n} \sum_{j=1}^n Y^j Y_u^j - \left(\frac{1}{n} \sum_{j=1}^n Y^j \right) \left(\frac{1}{n} \sum_{j=1}^n Y_u^j \right)}{\frac{1}{n} \sum_{j=1}^n (Y^j)^2 - \left(\frac{1}{n} \sum_{j=1}^n Y^j \right)^2} . \quad (3.5)$$

Note that this method can be applied to independent inputs or independent groups of inputs. Its main drawback is the increasing number of needed evaluations. Estimating all first-order, respectively all closed second-order, Sobol' indices requires $n(d+1)$, respectively $n\left(\binom{d}{2} + 1\right)$, evaluations. For some models where a single evaluation takes a few minutes to one hour, this solution becomes rapidly intractable in case of large input space dimension.

Some improvements have been introduced by Saltelli [Sal02] to reduce the number of evaluations but with a cost still depending of the dimension d of the input space. A solution to this issue lies in the use of the replication method as explained in the next section.

3.3 Replication method

The replication method has been described for first-order Sobol' indices estimation by Mara *et al.* [MJ08]. It has been combined with the use of randomized orthogonal array for closed second-order Sobol' indices estimation in Tissot and Prieur [TP15]. We propose hereafter a description of these methods and their generalization to independent groups of dependent variables.

3.3.1 Replication method for first-order Sobol' indices estimation

In their paper [MJ08], Mara *et al.* propose a comparison between a deterministic sensitivity analysis and a stochastic sensitivity analysis. The deterministic approach is achieved using the improved FAST method. The stochastic methodology is performed with a modification of the Sobol' method using the concept of replication.

This improved Sobol' method consists in two replicated Monte Carlo designs in the case of d independent inputs. As described below, this method can be extended for groups of independent inputs.

Let us consider p independent one-dimensional or multidimensional input variables $X_1, \dots, X_I, \vec{X}_{I+1}, \dots, \vec{X}_p$ as in (3.4). Thus $p = I + m$.

Let
$$D = \{X^j = (X_1^j, \dots, X_I^j, \vec{X}_{I+1}^j, \dots, \vec{X}_p^j), 1 \leq j \leq n\}$$

be a design where X_i^j denotes the j -th sample of X_i . Each X^j is therefore a point in the d -dimensional input space. We say that a design D' is replicated from D if it is obtained through a column-wise permutation of D . More precisely it means that :

$$D' = \{X'^j = (X_1^{\pi_1(j)}, \dots, X_I^{\pi_I(j)}, \vec{X}_{I+1}^{\pi_{I+1}(j)}, \dots, \vec{X}_p^{\pi_p(j)}), 1 \leq j \leq n\}$$

where the vectors π_i $i \in \{1, \dots, p\}$ are p independent random variables uniformly distributed in the set of permutations Π_n and $\pi_i(j) = \pi_i^j$ denotes the j -th component of π_i .

Let us denote by f the model and by Y_D and $Y_{D'}$ the two vectors of model outputs associated to D and D' :

$$Y_D = \{Y_D^j = f(X_1^j, \dots, X_I^j, \vec{X}_{I+1}^j, \dots, \vec{X}_p^j), 1 \leq j \leq n\} ,$$

$$Y_{D'} = \{Y_{D'}^j = f(X_1^{\pi_1(j)}, \dots, X_I^{\pi_I(j)}, \vec{X}_{I+1}^{\pi_{I+1}(j)}, \dots, \vec{X}_p^{\pi_p(j)}), 1 \leq j \leq n\} .$$

To estimate the index S_k , $k \in \{1, \dots, p\}$, the values of Y_D are rearranged with the corresponding permutation π_k . As a result, it looks like $Y_{D'}$ has been constructed by varying all groups of inputs except the k -th. This is the same concept of "freezing" as the one introduced by Sobol' [Sob93].

Then using (3.5) with Y_D rearranged and $Y_{D'}$ in place of Y and Y_I we obtain $\widehat{S}_k$ (an illustration of the method is proposed in the appendix 3.A in the case of independent inputs).

3.3.2 Replication method for closed second-order Sobol' indices estimation

In the case of closed second-order Sobol' indices estimation, the replication method does not consist anymore into a column-wise set of permutations. Indeed, to estimate closed second-order Sobol' indices, a way to "freeze" each subset of two variables has to be found. We need to find a structure allowing to get the same 2-sets of values in each subset of two columns of each replicated design. Such a structure has been introduced by Kishen [Kis42] and further extended by Rao [Rao46] and is known as orthogonal array. It is defined as follows :

Definition 3.1. A $t - (q, d, \lambda)$ orthogonal array ($t \leq d$) is a $\lambda q^t \times d$ matrix whose entries are chosen from a q -set of $\mathbb{N}$ such that in every subset of t columns of the array, every t -subset of points of this q -set appears in exactly λ rows.

From this definition by setting $t = 2$, we can construct a structure consisting of points in $\{1, \dots, q\}^{\lambda q^2}$ where each 2-set of columns have the same 2-set of points λ times.

We present now a result from the method of differences introduced by Bose and Bush [BB52] to construct a $2 - (q, p, 1)$ linear orthogonal array. We begin with some useful definitions and a theorem. This theorem is a consequence of the method of differences [BB52, Theorem 3], the proof can be found in [HSS99, Lemma 6.12] :

Definition 3.2. *A linear orthogonal array is a $t - (q, d, \lambda)$ orthogonal array where $\lambda = 1$ and q is a prime number. The set of rows of a linear orthogonal array is a subspace of $(\mathbb{Z}/q\mathbb{Z})^d$.*

Definition 3.3. *Let G be the Galois field of order q , $GF(q)$, a $\lambda q \times d$ matrix $D(\lambda q, d, q)$ is a difference matrix (or difference scheme) of strength 2 if for every subset of 2 columns of the matrix, the vector of row-wise difference of the subset contains every elements of $GF(q)$ λ times.*

Theorem 3.1. *Let M be a $D(\lambda q, d, q)$ difference matrix of strength 2, B the vector of elements of $GF(q)$, $\oplus$ the Kronecker sum and T the transpose operator then $(M \oplus B)^T$ is a $2 - (q, d, \lambda)$ orthogonal array.*

Now, consider q a prime number and $G = GF(q)$ the Galois field associated, the multiplication table of G is a $D(q, q, q)$ difference matrix of strength 2 with $\lambda = 1$. From Theorem 3.1 we can construct a $2 - (q, p, 1)$ linear orthogonal array : $A = (M \oplus B)^T$ where M is a sub-matrix of p columns of the multiplication table of G where p still denotes the number of groups of independent inputs. What is left is to connect this $2 - (q, p, 1)$ orthogonal array to the replication method. We consider a $q \times p$ design :

$$D = \{X^j = (X_1^j, \dots, X_I^j, \vec{X}_{I+1}^j, \dots, \vec{X}_p^j), 1 \leq j \leq q\}$$

where $X_1, \dots, X_I, \vec{X}_{I+1}, \dots, \vec{X}_p$ are one-dimensional or multidimensional independent variables and q is a prime number. Denote by A the previous $2 - (q, p, 1)$ orthogonal array. We can establish a direct bijection between the indices $\{1, \dots, q\}$ of rows of D and the set $\{1, \dots, q\}$ from which A is constructed. Each column of A will serve as new indices to build the respective column of the new design. Resulting from the construction procedure, in each column of A , each entry of $\{1, \dots, q\}$ appears q times. This is due to the fact that each entry is involved in q different 2-sets. The number of rows of A is equal to q^2 meaning that using the previous bijection we can construct a design D_{OA} of q^2 rows where each column has the same entries of the corresponding column of D but re-indexed from the elements of the corresponding column of A :

$$D_{OA} = \{X^j = (X_1^{A_1^j}, \dots, X_I^{A_I^j}, \vec{X}_{I+1}^{A_{I+1}^j}, \dots, \vec{X}_p^{A_p^j}), 1 \leq j \leq q^2\}.$$

This gives us the first of the two needed replicated designs. To replicate D_{OA} we consider once again random vectors of permutations π_i in Π_q . We also have a direct

bijection between the elements $\{1, \dots, q\}$ of each π_i and the set $\{1, \dots, q\}$ from which A is constructed. This allows the following construction of the second needed replicated design :

$$D'_{OA} = \{X'^j = (X_1^{\pi_1(A_1^j)}, \dots, X_I^{\pi_I(A_I^j)}, \vec{X}_{I+1}^{\pi_{I+1}(A_{I+1}^j)}, \dots, \vec{X}_p^{\pi_p(A_p^j)}), 1 \leq j \leq q^2\}$$

where each π_i is re-indexed from the elements of the corresponding column of A . In fact, the matrix whose columns are the vectors $\pi_1(A_1), \dots, \pi_p(A_p)$ is also an orthogonal array.

We obtain two designs of length q^2 . Denote then $Y_{D_{OA}}$ and $Y_{D'_{OA}}$ the two vectors of model outputs associated to D_{OA} and D'_{OA} . To get the same "freezing" concept as in the last subsection, but this time for each subset of two coordinates $(k, l) \in \{1, \dots, p\}^2$, we apply the following transformation :

$$\forall j \in \{1, \dots, q^2\} : Y^{A_k^j + A_l^j * q - q} = Y_{D_{OA}}^j, Y'^{\pi_k(A_k^j) + \pi_l(A_l^j) * q - q} = Y_{D'_{OA}}^j.$$

As a result, Y and Y' have been constructed by varying all groups of inputs except the k -th and l -th. This allows us to estimate the index $\underline{S}_{kl}$ with the formula (3.5) by replacing Y_I with Y' (an illustration of the method is proposed in the appendix 3.B in the case of independent inputs).

Remark 3.1. *The construction of a $2 - (q, d, 1)$ linear orthogonal array is possible only if $q \geq d - 1$. The total number of evaluation required by the replication method will be $2q^2$.*

Remark 3.2. *For the sake of clarity, we have treated the case of first-order and closed second-order indices estimation separately. Actually, given that a $1 - (q, d, 1)$ orthogonal array is a matrix of permutations we could have written a general definition for both cases.*

Remark 3.3. *When the inputs are independent, the former definitions of the two replicated designs are simplified. For the case of first-order indices we obtain :*

$$D = \{X^j = (X_1^j, X_2^j, \dots, X_d^j), 1 \leq j \leq n\},$$

$$D' = \{X'^j = (X_1^{\pi_1(j)}, X_2^{\pi_2(j)}, \dots, X_d^{\pi_d(j)}), 1 \leq j \leq n\}.$$

which are exactly the designs used in Mara et al. [MJ08]. For the case of closed second-order indices we obtain :

$$D_{OA} = \{X^j = (X_1^{A_1^j}, X_2^{A_2^j}, \dots, X_d^{A_d^j}), 1 \leq j \leq q^2\},$$

$$D'_{OA} = \{X'^j = (X_1^{\pi_1(A_1^j)}, X_2^{\pi_2(A_2^j)}, \dots, X_d^{\pi_d(A_d^j)}), 1 \leq j \leq q^2\}.$$

where A is a $2 - (q, d, 1)$ orthogonal array. The design D_{OA} above generalizes the randomized orthogonal arrays used in the paper by Tissot et al. [TP15] (see also Owen [Owe94] or Lemieux [Lem09] for the definition).

3.4 Sampling strategy under linear ordered constraints

In the previous section a method based on replicated designs to estimate Sobol' indices for independent groups of inputs was proposed. The fact that it handles groups of inputs is a way to tackle the problem of dependence within parameters. Indeed, each set of dependent inputs is treated as a group. The first- and second-order indices are calculated for the group, and not independently for each parameter (which would provide values that can not be interpreted easily). This method can be used for any kind of dependency, under the condition that one is able to properly sample from the input space; samples that are further needed to feed the designs. When inputs are independent, getting samples uniformly distributed in $[0, 1]$ is sufficient. When dependency is expressed through constraints, one has to create samples that satisfy the constraints, which is not an easy task in general. As mentioned in the introduction, the work presented in this paper is motivated by a variance-based sensitivity analysis we want to conduct on a land use and transport integrated model, called *Tranus* [dlB99]. Such a model creates a numerical representation of the transport network on a territory, and calculates the spatial repartition of the economic activities. It is based on economic laws whose parameters can be linked through ordered linear constraints. We thus focus on such constraints and propose an adapted sampling strategy based on the simplex geometric structure. This strategy is further enhanced with a space filling methodology.

3.4.1 Sampling strategy based on the simplex to handle constraints

We consider the same vector of inputs X as defined in (3.4). Each unidimensional variable is uniformly distributed in $[0, 1]$. Each multidimensional variable $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$ contains k_i variables valued in $[0, 1]$ and is subject to the following linear ordered constraints :

$$X_{I+1+\Sigma_{i-1}} \leq X_{I+2+\Sigma_{i-1}} \leq \dots \leq X_{I+\Sigma_i} . \quad (3.6)$$

In case of absence of constraints, the joint distribution of each multidimensional variable $\vec{X}_{I+i}$ is a uniform distribution on the unit k_i -hypercube. The introduction of constraints (3.6) transforms the support of the joint uniform distribution in a k_i -simplex. The general definition of a simplex is the following :

Definition 3.4. *A simplex, or k -simplex, is the generalization of a triangle (2-simplex) or tetrahedron (3-simplex) to any dimension k . Let Δ_k be the simplex with vertices $P_0, \dots, P_k$ in $\mathbb{R}^k$:*

$$\Delta_k = \{M = \sum_{j=0}^k w_j P_j \mid w_j \geq 0, \sum_{j=0}^k w_j = 1\}$$

To sample from a uniform distribution over a simplex we refer to the following theorem of Devroye [Dev86] :

Theorem 3.2. *Let Δ_k be a simplex in $\mathbb{R}^k$ with vertices $P_0, \dots, P_k$ and $U_1, \dots, U_k$ be independent variables uniformly distributed in $[0, 1]$. We define $W_0, \dots, W_k$ the associated differences as follows :*

$$\forall j \in \{1, \dots, k\}, W_j = V_{j+1} - V_j, V_0 = 0, V_{k+1} = 1$$

where V_j denote the j -th ordered statistics constructed from the set $(U_i)_{i=1, \dots, k}$.

$$\forall j \in \{1, \dots, k\}, V_j = \min_{i=1, \dots, k} \left\{ U_i \setminus \bigcup_{l=1}^{j-1} V_l \right\}.$$

Then the random variable $Z = \sum_{j=0}^k W_j P_j$ is uniformly distributed in the simplex Δ_k .

To each set of constraints (3.6) is associated a k_i -simplex, defined as follows :

$$\Delta_{k_i}^* = \left\{ M = \sum_{j=0}^{k_i} w_j e_j^* \mid w_j \geq 0, \sum_{j=0}^{k_i} w_j = 1 \right\}$$

with vertices the ordered vectors of the following basis of $\mathbb{R}^{k_i}$:

$$e_0^* = (0, 0, \dots, 0, 0), e_1^* = (0, 0, \dots, 0, 1), e_2^* = (0, 0, \dots, 1, 1), \dots, e_{k_i}^* = (1, 1, \dots, 1, 1)$$

Since the vertices of $\Delta_{k_i}^*$ satisfy ordered linear constraints (3.6), then every point sampled in $\Delta_{k_i}^*$ will also satisfy those constraints. Sampling in $\Delta_{k_i}^*$ using the strategy of Theorem 3.2 is equivalent to sampling uniformly under a set of k_i linear ordered constraints (3.6).

This method enables to sample each of the m multidimensional variables contained in the vector of inputs X . For the remaining independent one-dimensional variables we use uniform random numbers. Algorithm 3.1 summarizes the whole sampling strategy to get n samples :

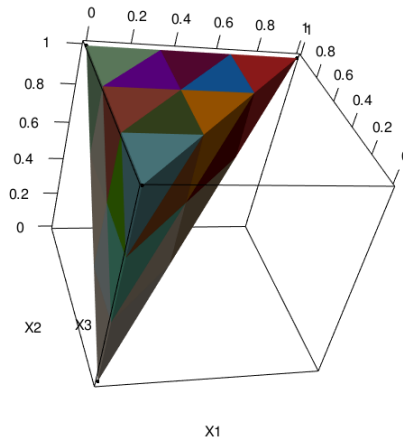
Algorithm 3.1 Sampling n points from d inputs under m disjoint k_i -sets of linear ordered constraints

1. for each of the m multidimensional variables $\vec{X}_{I+i}$
 - create the k_i -simplex
 - sample n points with Theorem 3.2
 2. for each of the $d - \sum_{i=1}^m k_i$ one-dimensional variables
 - sample n random uniform numbers.
-

3.4.2 Space filling strategy

In the classical case of independent variables, one should prefer to use a space filling sampling (Latin Hypercube or low discrepancy sequence) rather than sampling each variable from a uniform distribution. Generally, such a procedure ensures that the input space will be properly explored. To ensure the same property in our case of constrained input space, the proposed sampling method has to be improved as nothing prevents all the points to be located in a small area of the space. A way to do so is to first subdivide each k_i -simplex into partitions of sub-simplices of order k_i and then sample uniformly from the latter using Theorem 3.2. Hence, the subdivision method can be interpreted as a simplex version of the Latin Hypercube sampling. Figure 3.1 shows an example of the subdivision obtained on the 3-simplex Δ_3^* . Algorithm 3.2 summarizes the subdivision method (an illustration of the algorithm

FIGURE 3.1 – subdivision for the 3-simplex Δ_3^* with $n = 27$



is proposed in the appendix 3.C for a 2-simplex). For the sake of clarity, in the following, k_i has been replaced by k .

In the next subsection, we compare the equidistribution properties of the subdivision method.

3.4.3 Equidistribution properties

Quasi-periodic distributions are known to offer a good compromise between uniformity and regularity properties. In the sensitivity analysis framework, no optimal design based on uniformity or regularity properties has yet been proposed. As such, we aim to reach a compromise between both properties with our subdivision method by creating point sets close to those obtained from quasi-periodic distributions. Three criteria were selected to measure those properties : the maximin, the emst and the simplex star discrepancy [PC04].

The maximin criterion can be interpreted as follows : the higher the value, the

Algorithm 3.2 Space filling sampling of n points for a set of k linear ordered constraints

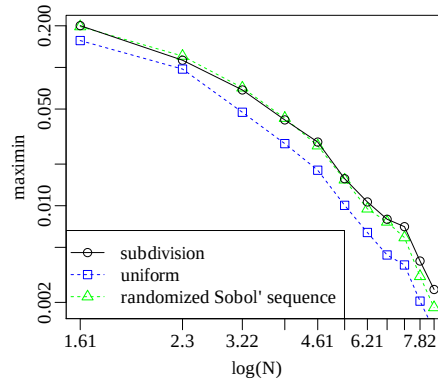
1. create the k^2 vertices of the unit k -hypercube and assign a number to each vertex according to the order of creation
 2. identify the $k + 1$ vertices whose coordinates satisfy the set of constraints
 3. store those vertices into a $(k + 1) \times k$ matrix M_{coord} , each column corresponding to one of the k coordinates
 4. create the $k!$ vectors of permutations of Π_k
 5. apply each permutation to the columns of M_{coord} , this creates a total of $k!$ matrices
 6. for each one of the $k!$ matrices, match the vertices with those of the unit k -hypercube and retrieve the vector of matching numbers, this creates a total of $k!$ vectors of matching numbers that we note ω_j , $j \in \{1, \dots, k!\}$
 7. choose $\alpha = \text{floor}(n^{\frac{1}{k}})$ (respectively $\alpha = n^{\frac{1}{k}}$ if $n^{\frac{1}{k}} \in \mathbb{N}$) the number of levels
 8. subdivide the unit k -hypercube into $(\alpha + 1)^k$ (respectively α^k if $n^{\frac{1}{k}} \in \mathbb{N}$) sub-hypercubes. For each sub-hypercube, assign a number to each vertices following the same order of numeration as in 1.
 9. for each sub-hypercube :
 - for each ω_j , conserve the vertices whose indices correspond to ω_j , this gives $k!$ sets of vertices
 - among those sets keep only those whose coordinates satisfy the constraints
 10. randomly discard $(\alpha + 1)^k - n$ of all the sets created to keep only n sets
 11. sample one point from each set (each set are the vertices of a simplex) using Theorem 3.2 to get a total of n points.
-

more regular the scattering of design points. The emst criterion can be interpreted using a (μ, σ) graph, first graph of Figure 3.3, called interpretation graph. The uniform distribution is used as a reference. μ stands for the mean of the edges lengths, σ for the standard deviation of the edges lengths. A value of the emst criterion is represented as a point in the (μ, σ) graph. Maximin and emst (Euclidean minimal spanning tree) criteria provide together a good estimation of the regularity properties of a design.

The simplex star discrepancy criterion measures the uniformity property of a point set. We propose to compare three samples on the simplex based on those three criteria : (a) uniform sample created with Algorithm 3.1, (b) sample generated from the "Sort" method proposed in [PC04], that consists in sorting each point, coordinate-wise, of a Sobol' sequence, that has been randomized by applying a Cranley-Patterson rotation (c) sample generated with the subdivision method.

The results are averaged over $r = 100$ repetitions. As stated in [PC04], the algorithm used to compute the simplex star discrepancy is inadequate for dimension higher than 3. Thus, we compare the three samples on the simplex Δ_3^* . Results for the maximin criterion are shown on Figure 3.2. Looking at the results, the subdivision

FIGURE 3.2 – Log-log graph of averaged maximin criterion over $r = 100$ repetitions on Δ_3^* for different size of design N .



design and the randomized Sobol' sequence design give similar results. Those two designs are more regular than the uniform design.

The results for the emst criterion are shown on Figure 3.3. We compute $r = 100$ repetitions of the emst criterion for each of the three designs. Using the interpretation graph as a reference, the subdivision design and the randomized Sobol' sequence design look closer to quasi-periodic distributions than uniform distributions. The subdivision method is the most regular.

The results for the simplex star discrepancy criterion are shown on Table 3.1. We used the same program as in [PC04] written by E. Thiémarc [Thi04]. We adapted this program for the simplex. The program returns an upper and a lower bound for the discrepancy. Looking at the table, the randomized Sobol' sequence performs the best for this criterion. The subdivision gives results closer to those obtained with

FIGURE 3.3 – averaged emst criterion over $r = 100$ repetitions on Δ_3^* for three size of design N and interpretation graph on the topleft.

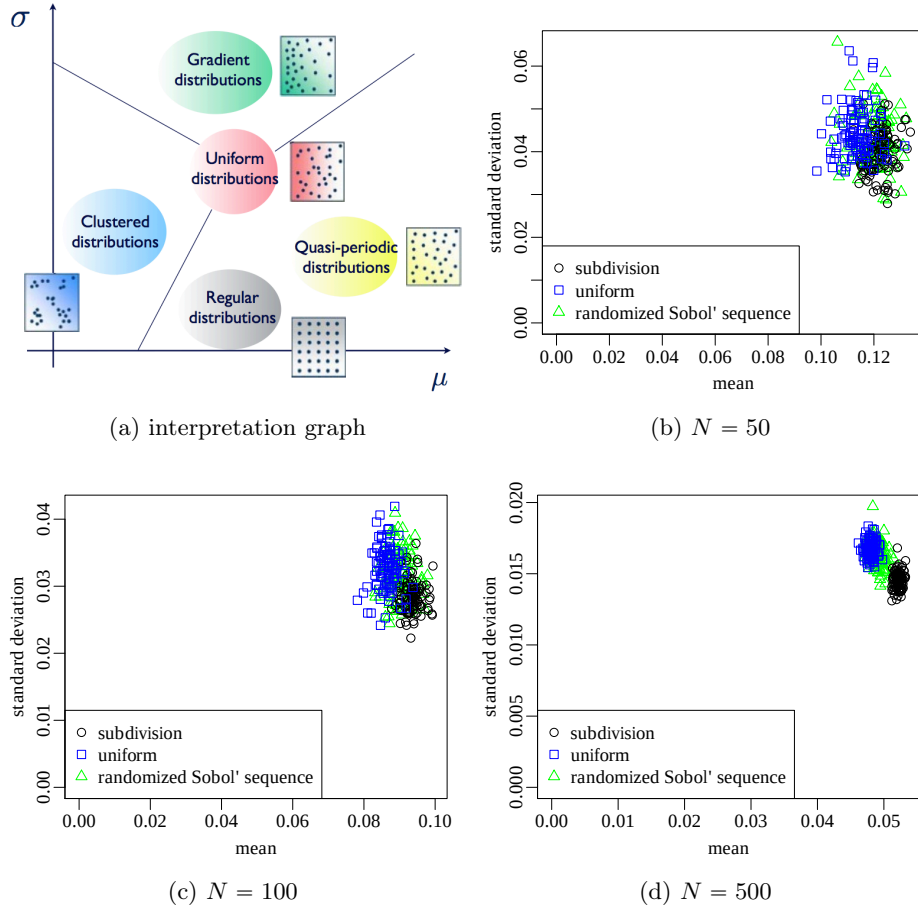


TABLE 3.1 – averaged bounds for the simplex star discrepancy criterion over $r = 100$ repetitions on Δ_3^* for different size N of the three designs : randomized Sobol' sequence, subdivision, uniform. L stands for lower, U for upper.

N	bounds	10	100	1000	5000	10000
Randomized	L	0.3172	0.0790	0.0140	0.0045	0.0025
Sobol' sequence	U	0.3403	0.1063	0.0526	0.0490	0.0485
Subdivision	L	0.3429	0.0973	0.0214	0.0102	0.0059
	U	0.3659	0.1224	0.0555	0.0497	0.0489
Uniform	L	0.3863	0.1318	0.0421	0.0191	0.0129
	U	0.4099	0.1568	0.0706	0.0534	0.0506

the randomized Sobol' sequence than with the uniform design.

In this section, uniformity and regularity properties of three designs have been compared on the base of three criteria. For the design created from our subdivision method, a good compromise is obtained between uniformity and regularity properties. In Section 3.6.1, the same designs are used within the replication method to assess their respective influence on the estimation of Sobol' indices.

3.5 Summary of the subdivision method

Consider the vector of inputs :

$$X = (X_1, \dots, X_I, \underbrace{X_{I+1}, \dots, X_{I+k_1}}_{\vec{X}_{I+1}}, \dots, \underbrace{X_{I+1+\Sigma_{i-1}}, \dots, X_{I+\Sigma_i}}_{\vec{X}_{I+i}}, \dots, \underbrace{X_{I+k_1+k_2+\dots+k_{m-1}+1}, \dots, X_d}_{\vec{X}_{I+m}}) \text{ where } \Sigma_i = \sum_{l=0}^i k_l, \quad k_0 = 0.$$

Each multidimensional variable $\vec{X}_{I+i}$, $i \in \{1, \dots, m\}$ contains k_i variables and is subject to the following linear ordered constraints :

$$X_{I+1+\Sigma_{i-1}} \leq X_{I+2+\Sigma_{i-1}} \leq \dots \leq X_{I+\Sigma_i}.$$

Let d be the dimension of input space, $K = \sum_{i=1}^m k_i$ be the number of constrained inputs, m be the number of sets of constrained variables then the number of grouped Sobol' indices to estimate is $p = d - K + m$ for first-order indices and $\binom{p}{2}$ for closed second-order indices.

Let $(A_i^j)_{i=1..p, j=1..n}$ be an orthogonal array of strength t . $t = 1$ respectively 2 to estimate first- respectively second-order indices. Let $n = q^t \in \mathbb{N}^*$ be the number of samples of the input space with q a prime number if $t = 2$. Denote $(D_{OA_i}^j)_{i=1, \dots, d, j=1, \dots, n}$ and $(D'_{OA_i}^j)_{i=1, \dots, d, j=1, \dots, n}$ the two replicated designs.

1. For the independent inputs X_i $i \in \{1, \dots, I\}$, construct the replicated designs sampling from Latin Hypercube [MBC79] to get space filling uniform distribution.

$$D_{OA_i}^j = \frac{A_i^j - U_i^{A_i^j}}{q},$$

$$D'_{OA_i}^j = \frac{\pi_i(A_i^j) - U_i^{\pi_i(A_i^j)}}{q}, \quad i \in \{1, \dots, I\}.$$

where the π_i and the U_i^j are independent random variables uniformly distributed in Π_n and $[0, 1]$.

2. For the constrained inputs $\vec{X}_{I+i}$ $i \in \{1, \dots, m\}$, construct the replicated designs sampling from algorithm 3.2 to get a space filling uniform distribution in the corresponding ordered k_i -simplices.

$$D_{OA}^j_{I+1+\Sigma_{i-1}, I+2+\Sigma_{i-1}, \dots, I+\Sigma_i} = \sum_{l=0}^{k_i} W_l^{A_i^j} P_l^{A_i^j},$$

$$D'_{OA}^j_{I+1+\Sigma_{i-1}, I+2+\Sigma_{i-1}, \dots, I+\Sigma_i} = \sum_{l=0}^{k_i} W_l^{\pi_i(A_i^j)} P_l^{\pi_i(A_i^j)}, \quad i \in \{1, \dots, m\}.$$

where $W_0^{A_i^j}, \dots, W_{k_i}^{A_i^j}$ (respectively $W_0^{\pi_i(A_i^j)}, \dots, W_{k_i}^{\pi_i(A_i^j)}$) are the associated differences constructed from $U_{I+1+\Sigma_{i-1}}^{A_i^j}, \dots, U_{I+\Sigma_i}^{A_i^j}$ (respectively $U_{I+1+\Sigma_{i-1}}^{\pi_i(A_i^j)}, \dots, U_{I+\Sigma_i}^{\pi_i(A_i^j)}$) and $P_0, \dots, P_{k_i}$ are the columns of the $n \times (k_i + 1)$ matrix P storing the sets of vertices resulting from Algorithm 3.2.

3. Denote then $Y_{D_{OA}}$ and $Y_{D'_{OA}}$ the two vectors of model outputs associated to D_{OA} and D'_{OA} we compute the Sobol' indices with the methodology presented in section 3.3.

Remark 3.4. *As we stated at the beginning of section 3.4, we can apply the replication method to any kind of distributions under any type of correlation. The only requirement is to be able to generate a design or in other words to be able to sample each multidimensional variable (see section 3.6 for a multivariate Gaussian example).*

3.6 Application to test functions

3.6.1 Influence of designs on the replication method

In this section, the replication method is applied with each of the three designs introduced in Section 3.4.3 : (a) uniform, (b) randomized Sequence Sobol', (c) subdivision. We compare the results obtained with each design on the estimation of first-order and closed second-order Sobol' indices. We use the estimator with formula given in (3.5). The estimation procedure is repeated $r = 1000$ times to get a sample of estimated indices ; and this for various values of size N of the design. To evaluate the precision, we calculate the mean square error defined as :

$$MSE_N = \frac{1}{N} \sum_{k=1}^r (\hat{S}_{k,N} - S)^2,$$

where $\hat{S}_{k,N}$ is the k -th estimation of the index, and S the theoretical value of the index. Since f has an analytical expression, Sobol' indices can be calculated by

symbolic integrals to evaluate their theoretical value. We consider the g-function introduced by Sobol' [Sob93]. The g-function is defined as follows :

$$f(X_1, \dots, X_d) = \prod_{i=1}^d g_i(X_i), \quad (3.7)$$

where

$$g_i(X_i) = \frac{|4X_i - 2| + a_i}{1 + a_i}, \quad a_i \geq 0.$$

Each value a_i determines the relative importance of the X_i . When the value of a_i gets closer to zero the variable X_i becomes more influential. Here, the values of the a_i are the following : ($a_1 = 1, a_2 = 2, \dots, a_7 = 7$). We consider the following sets of constraints :

- $X_1 \leq X_2$ and $X_3 \leq X_4$
- $X_5 \leq X_6 \leq X_7$

The results shown in Figure 3.4 suggest that using either design (b) or (c) give overall better results than using the uniform design (a). For first-order indices, designs (b) and (c) give similar results. For closed-second order indices design (c) gives better results than (b) for the first values of N .

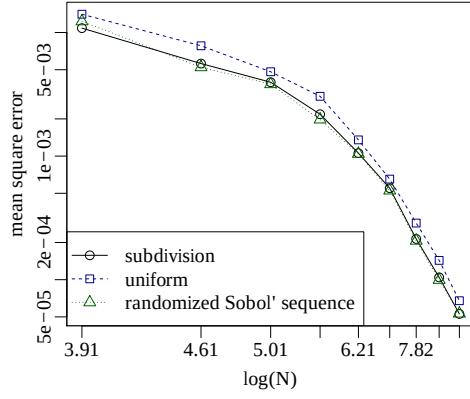
3.6.2 Replication method versus classical methods

In this section the replication method is applied to standard test functions with sets of linear ordered constraints. The design is generated with the subdivision method. The replication method is also applied to a case of a multivariate normal distribution. We compare the results to those given by the Saltelli method and the standard method of Sobol' that we both generalized to the case of groups of inputs (a first generalization of the standard method of Sobol' can be found in [JLD06]). For both the standard method and our replication method we use again the efficient estimator with formula given in (3.5). The size N of the design of experiments (DoE) is different for each method. Let n denote the numbers of points in the d -dimensional input space :

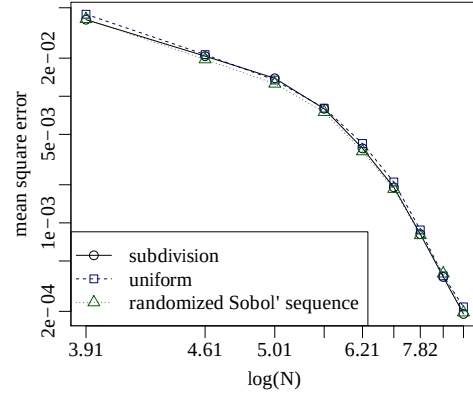
- for the replication method we get $N = 2n$ (each of the two replicated designs has n points) for first-order indices and $N = 2n$ for closed second-order Sobol' indices where $\sqrt{n}$ has to be a prime number
- for Saltelli's method we get $N = n(p + 2)$ for first-order indices and $N = n(2p + 2)$ for closed second-order indices where p is the number of independent one-dimensional or multidimensional input variables
- for the standard method we get $N = n(p + 1)$ for first-order indices and $N = n\left(\binom{p}{2} + 1\right)$ for closed second-order indices

From the user point of view, the total number of evaluations required is one of the most important aspects of the method. Thus, the comparison are made for similar computational time. For each method, we chose n to get the same value N or to approximate N as close as possible.

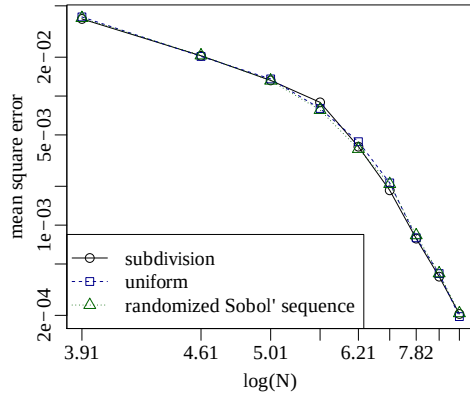
FIGURE 3.4 – Log-log graph of MSE_N criterion for first and closed second-order Sobol' indices given by the four methods for $r = 1000$ repetitions.



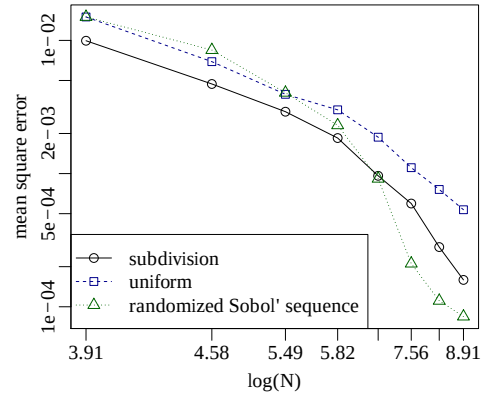
(a) Sobol' index $S_{1,2}$



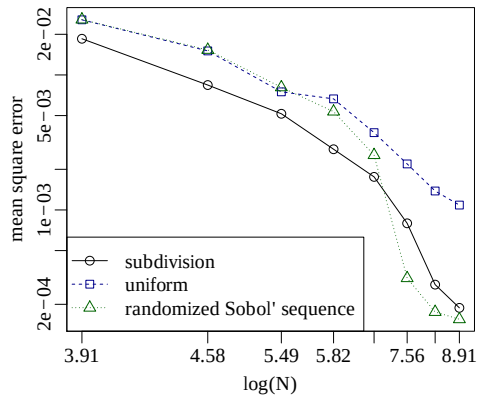
(b) Sobol' index $S_{3,4}$



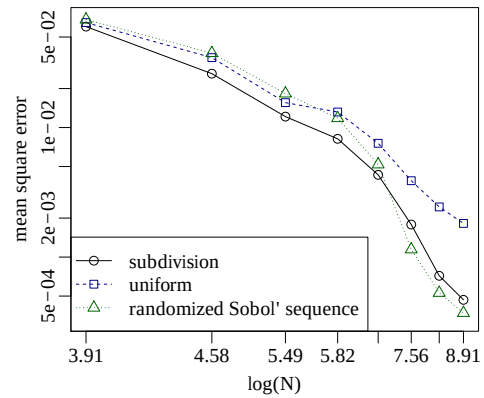
(c) Sobol' index $S_{5,6,7}$



(d) Sobol' index $\underline{S}_{\{1,2\}\{3,4\}}$



(e) Sobol' index $\underline{S}_{\{1,2\}\{5,6,7\}}$



(f) Sobol' index $\underline{S}_{\{3,4\}\{5,6,7\}}$

3.6.2.1 Sobol's g-function

We consider once again the g-function (3.7) as a test function. We choose here $d = 4$ and $(a_1 = 0, a_2 = 1, a_3 = 3, a_4 = 6)$. We consider the following sets of characteristics :

- $X_3 \leq X_4$ with (X_3, X_4) uniformly distributed in Δ_2^*
- X_1 and X_2 are independent and both uniformly distributed in $[0, 1]$.

Each of the three methods are used to compute the three first-order Sobol' indices and the three closed second-order Sobol' indices. The estimation procedure is repeated $r = 100$ times to get a sample of estimated indices ; and this for various values of N .

To evaluate the precision of each method, are shown for each index :

- boxplots drawn depending on increasing values of N ;
- evolutions of the error relative to the true value, depending on N . The error calculated is a variation of the *SAE* (Sum of Absolute Error) indicator used in [MJ08], and is defined as :

$$SAE_N = \sum_{k=1}^r \left| \hat{S}_{k,N} - S \right|,$$

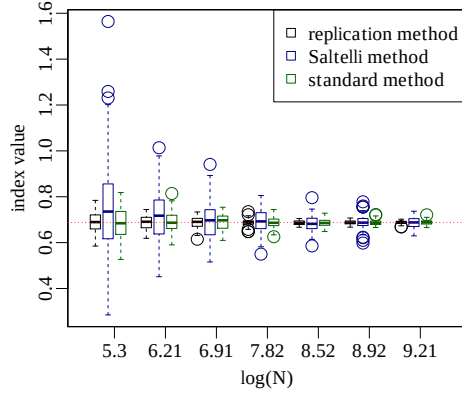
where $\hat{S}_{k,N}$ is the k -th estimation of the index, and S the theoretical value of the index.

Results on first-order indices are shown on Figure 3.5, while results on closed second-order Sobol' indices are shown on Figure 3.6. In both cases, the left column corresponds to boxplots and the right column to error curves. The theoretical values of indices are indicated by horizontal broken lines.

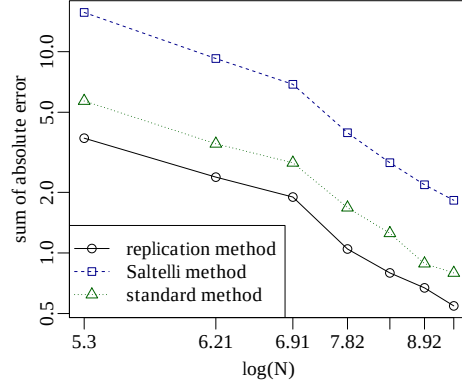
For the calculation of first-order indices, the observation is that the replication method gives better results, and particularly for low values of N . This observation is explained by the independence between the computational cost of our approach and the input space dimension. Thus, this observation is emphasized when we increase the number p of independent groups of inputs. The Saltelli's method gives better results when it comes to estimate indices with low value rather than indices with high value. At the opposite both the standard method and the replication method are better at estimating indices with high value but this could be due to the choice of the estimator rather than from the method itself (see Owen [Owe12] for a discussion on the estimation of small indices).

The replication method performs overall also better compared to the other two approaches for the calculation of closed second-order indices. The Saltelli method is once again better at estimating indices with low value and the other two methods are better at estimating indices with high value. Despite the difference, for low value indices our method still stays competitive.

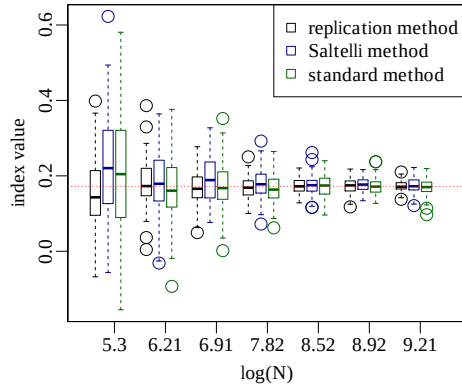
FIGURE 3.5 – Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions and for different size N of the design. Left column : boxplot representation. Right column : log-log graph of the SAE_N .



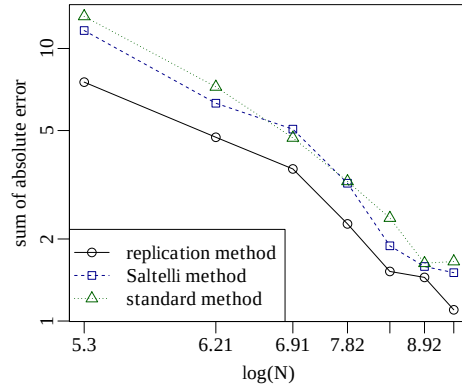
(a) Sobol' index S_1



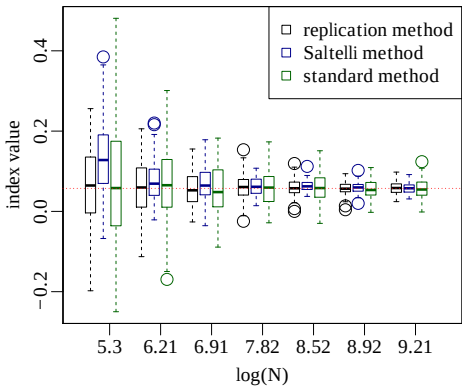
(b) Sobol' index S_1



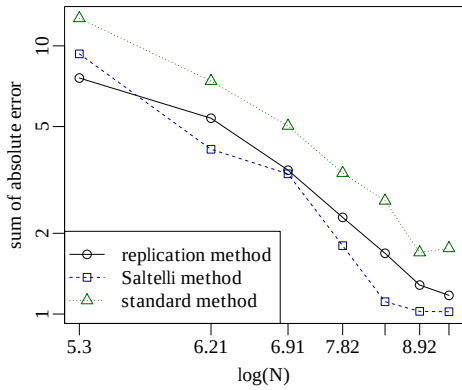
(c) Sobol' index S_2



(d) Sobol' index S_2

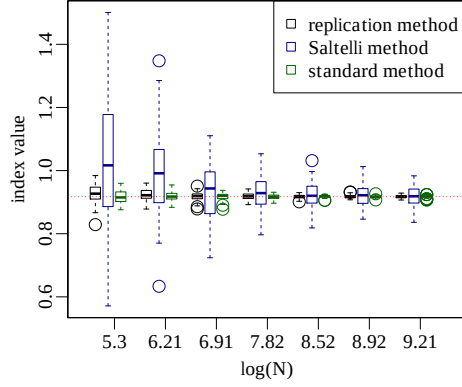
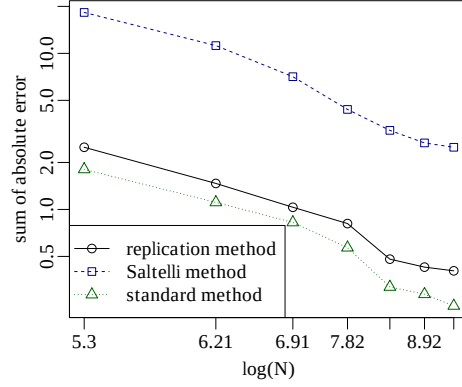
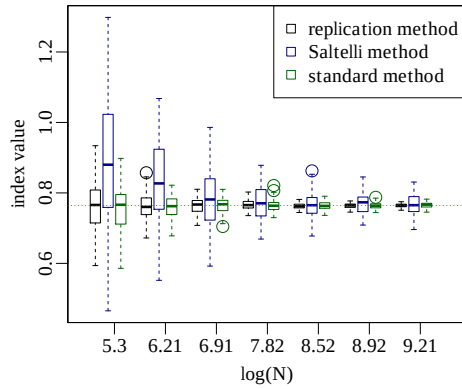
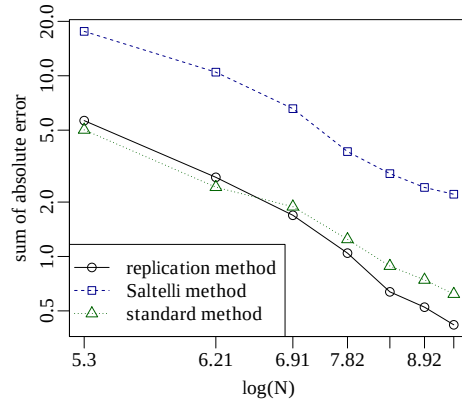
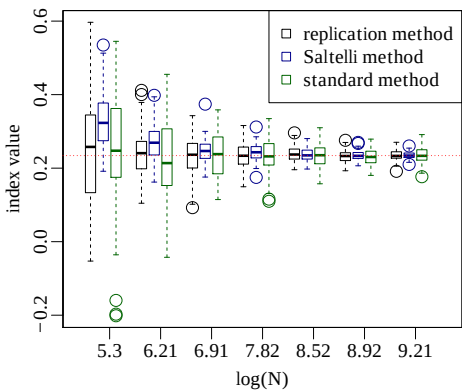
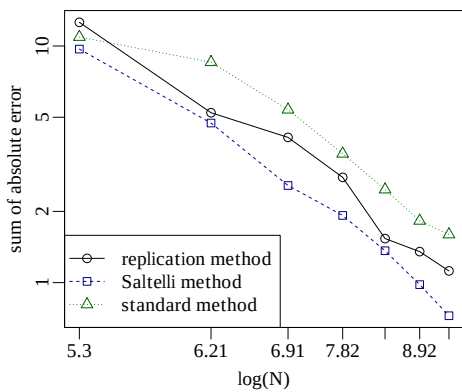


(e) Sobol' index $S_{3,4}$



(f) Sobol' index $S_{3,4}$

FIGURE 3.6 – Estimation of closed second-order Sobol' indices given by the three methods for $r = 100$ repetitions and for different size N of the design. Left column : boxplot representation. Right column : log-log graph of the SAE_N .

(a) Sobol' index $S_{1\{2\}}$ (b) Sobol' index $S_{1\{2\}}$ (c) Sobol' index $S_{1\{3,4\}}$ (d) Sobol' index $S_{1\{3,4\}}$ (e) Sobol' index $S_{2\{3,4\}}$ (f) Sobol' index $S_{2\{3,4\}}$

In order to test our method facing a high number of inputs, we consider the Sobol' g-function with a dimension of input space now equals to 50. The following characteristics are considered :

- $(X_1, X_{50}), (X_2, X_{49}), (X_3, X_{48}), (X_4, X_{47}), (X_5, X_{46})$ uniformly distributed in Δ_2^* ,
- $(X_6, X_{10}, X_{40}), (X_7, X_{11}, X_{39}), (X_8, X_{12}, X_{38}), (X_9, X_{13}, X_{37}), (X_{19}, X_{30}, X_{31})$ uniformly distributed in Δ_3^* ,
- $(X_{14}, X_{20}, X_{21}, X_{36}), (X_{15}, X_{22}, X_{23}, X_{35}), (X_{16}, X_{24}, X_{25}, X_{34}), (X_{17}, X_{26}, X_{27}, X_{33}), (X_{18}, X_{28}, X_{29}, X_{32})$ uniformly distributed in Δ_4^* ,
- $(X_{41}, X_{42}, X_{43}, X_{44}, X_{45})$ uniformly distributed in Δ_5^* .

The values of the a_i are the following : $(a_1 = \dots = a_{10} = 0, a_{11} = \dots = a_{20} = 1, a_{21} = \dots = a_{30} = 2, a_{31} = \dots = a_{40} = 4, a_{41} = \dots = a_{50} = 6)$.

Since there is a substantial number of indices, we choose not to draw all boxplots but instead to compute the two errors. The average of the p SAE_N calculated for each first-order index is referred as $ASAE_N^1$. The average of the $\binom{p}{2}$ SAE_N calculated for each closed second-order index is referred as $ASAE_N^2$.

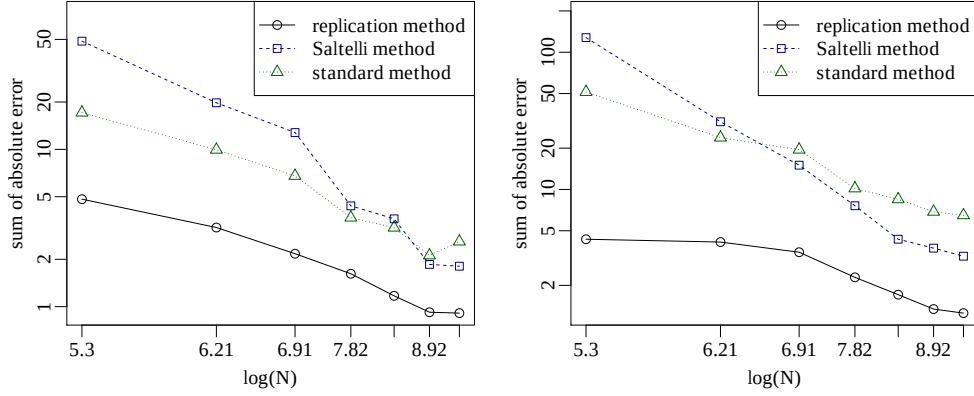
Table 3.2 gives the values of $ASAE_N^1$ for each method and each value of N . Figure 3.7 right represents the evolution curves of the $ASAE_N^2$. Since there is 120 closed second-order indices to estimate, the difference in that case between the results of the three methods is even more noticeable. Figure 3.7 left represents the evolution

TABLE 3.2 – Values of $ASAE_N^1$ for each method and for different size N of the design of experiments

size	replication	Saltelli	standard
$\log(N)$	method	method	method
5.30	4.82	48.74	17.12
6.21	3.19	19.78	9.94
6.91	2.17	12.79	6.78
7.82	1.62	4.38	3.68
8.52	1.17	3.62	3.18
8.92	0.92	1.86	2.12
9.21	0.91	1.81	2.59

curves of the $ASAE_N^1$ given N for the three methods. It clearly illustrates the benefit of the independence of the replication method's total cost to the dimension of the input space. Even for $N = 10^4$ the replication method is still in average two times better than the other two methods.

FIGURE 3.7 – Log-log graphs of $ASAE_N^1$ (left figure) and $ASAE_N^2$ (right figure) for each method depending on N



3.6.2.2 Bratley et al. Function

For the second example, we consider the function introduced by Bratley *et al.* [BFN92] The function is defined as follows :

$$f(X_1, \dots, X_d) = \sum_{i=1}^d (-1)^i \prod_{j=1}^i X_j .$$

We consider the following characteristics :

- $X_3 \leq X_4$ with (X_3, X_4) uniformly distributed in Δ_2^*
- X_1 and X_2 are independent and both uniformly distributed in $[0, 1]$.

We use the same estimation procedure as in the last subsection and we draw the same graphical results (Figure 3.8 and Figure 3.9).

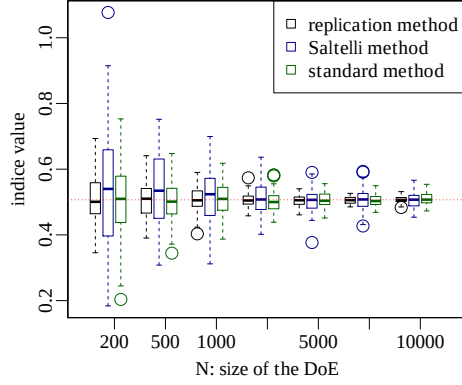
Conclusions are similar to the ones for the g-function of Sobol'. It shows the efficiency of our approach with respect to more standard approaches such as Saltelli or Sobol'.

3.6.2.3 Correlated multivariate normal distribution

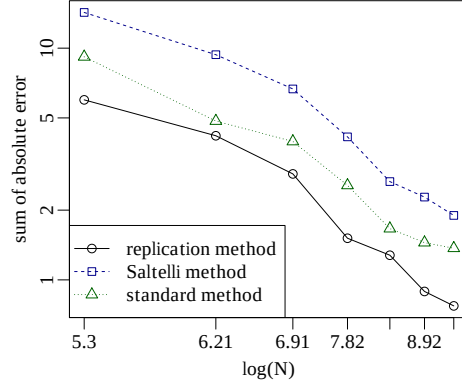
As stated in Remark 3.4, this example of a correlated multivariate normal distribution is to show that the replication method can be used in a general case of correlated inputs at the condition that we know how to sample each multidimensional input. Furthermore, the case of multivariate normal distribution with correlation terms is a common example of dependence for the inputs.

Let Σ be a positive-definite matrix, non necessarily diagonal. As in the previous subsection we write $X = (X_1, \dots, X_I, \vec{X}_{I+1}, \dots, \vec{X}_{I+m})$ where the multidimensional variables $\vec{X}_{I+j}$, $j \in \{1, \dots, m\}$ are independent random gaussian vectors with mean

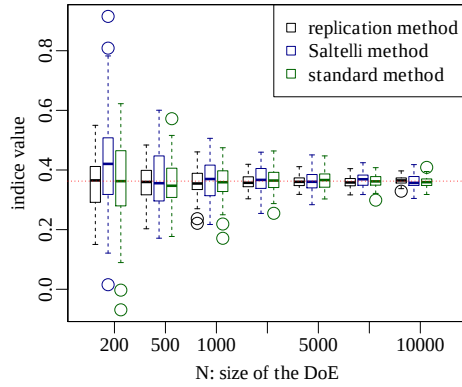
FIGURE 3.8 – Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions. Left column : boxplot representation for different values of N . Right column : log-log graph of the SAE_N for different values of N .



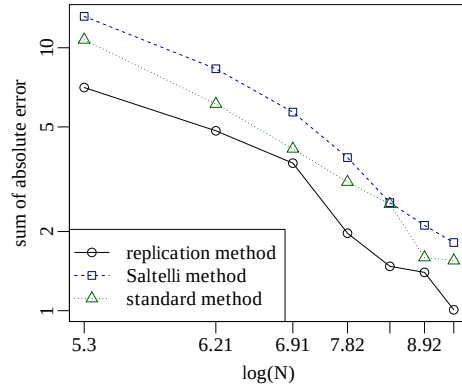
(a) Sobol' index S_1



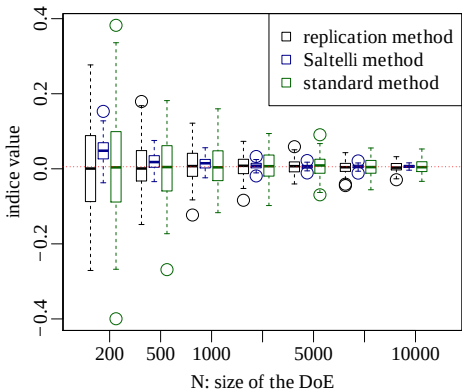
(b) Sobol' index S_1



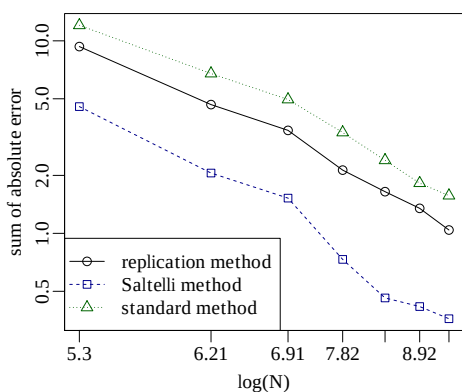
(c) Sobol' index S_2



(d) Sobol' index S_2

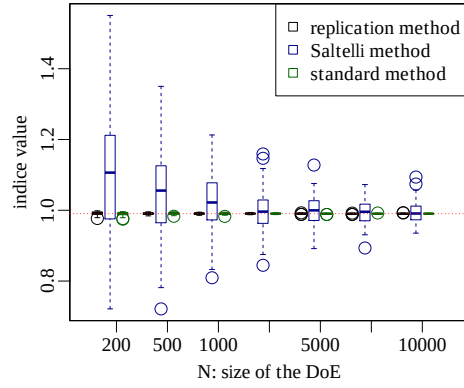
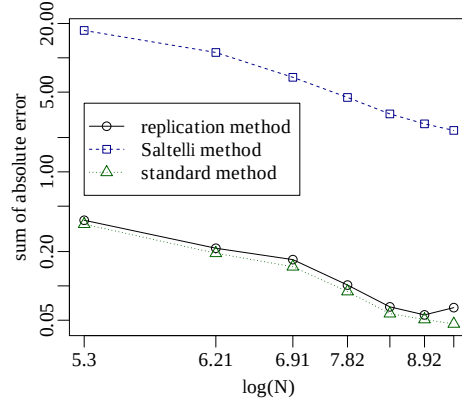
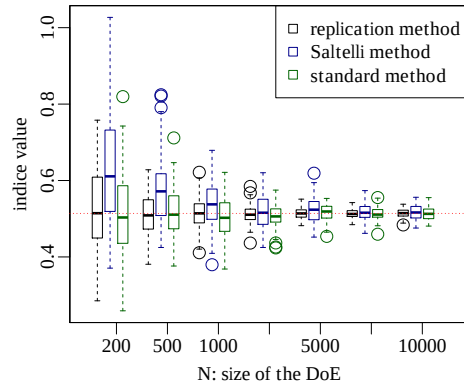
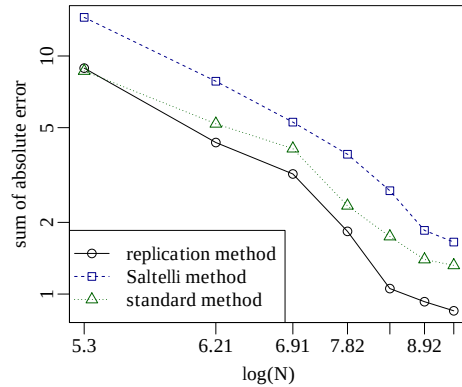
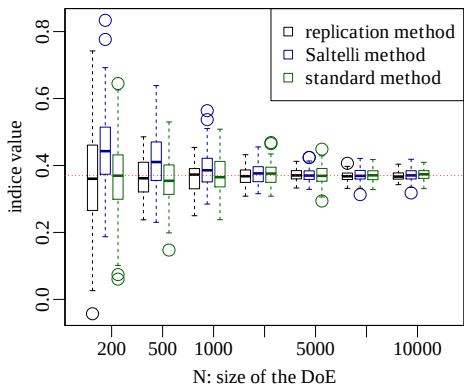
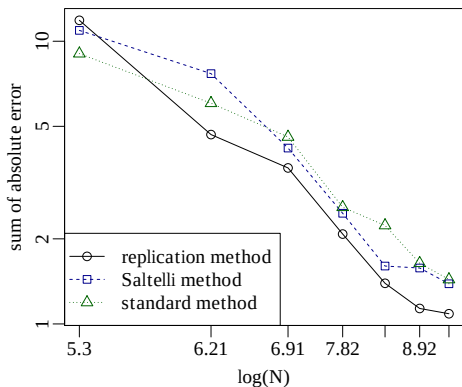


(e) Sobol' index $S_{3,4}$



(f) Sobol' index $S_{3,4}$

FIGURE 3.9 – Estimation of closed second-order Sobol' indices given by the three methods for $r = 100$ repetitions and for different size N of the design. Left column : boxplot representation. Right column : log-log graph of the SAE_N .

(a) Sobol' index $S_{1\{2\}}$ (b) Sobol' index $S_{1\{2\}}$ (c) Sobol' index $S_{1\{3,4\}}$ (d) Sobol' index $S_{1\{3,4\}}$ (e) Sobol' index $S_{2\{3,4\}}$ (f) Sobol' index $S_{2\{3,4\}}$

vector μ and covariance matrix Σ . Based on section 2.2 we can estimate the Sobol' indices for the groups of correlated variables $\vec{X}_{I+i}$.

To sample such a distribution we first use a Cholesky decomposition of the covariance matrix $\Sigma = LL^T$. Then, let $Z = (Z_0, \dots, Z_d)$ be a vector whose components are independent and follow a standard normal distribution. We obtain the true sampling from the formula : $X = \mu + LZ$. We obtain the two replicated designs of experiments D_{OA} and D'_{OA} with the formula from section 3 applied to our previous sample of X .

As an example we consider the following function inspired from the function presented in [LR11] :

$$f(X_1, X_2, X_3, X_4) = g_1(X_1, X_2) + g_2(X_2) + g_3(X_3) + g_4(X_4)$$

$$\text{where } \begin{cases} g_1(X_1, X_2) = (2(X_1 - \mu_1) + 1)(3(X_2 - \mu_2) + 2) \\ g_2(X_2) = 2(X_2 - \mu_2)^2 + X_2 - \mu_2 + 3 \\ g_3(X_3) = 1 + 2(X_3 - \mu_3) + 2(X_3 - \mu_3)^2 + 3(X_3 - \mu_3)^3 \\ g_4(X_4) = 1 + 4(X_4 - \mu_4) \end{cases}$$

and $X = (X_1, X_2, X_3, X_4)$ follows a centered multivariate normal distribution $\mathcal{N}(\mu, \Sigma)$ with parameters :

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho_{12}\sigma_1\sigma_2 & 0 & 0 \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{pmatrix} \quad \mu = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The first two variables X_1, X_2 are correlated and we chose the following values for the terms in Σ :

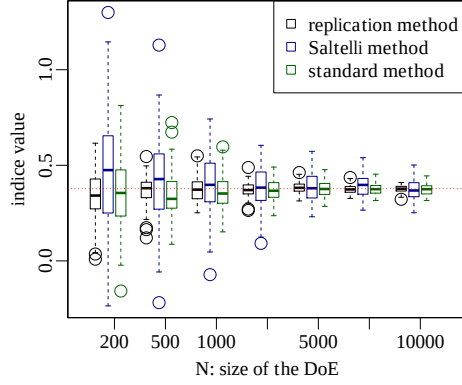
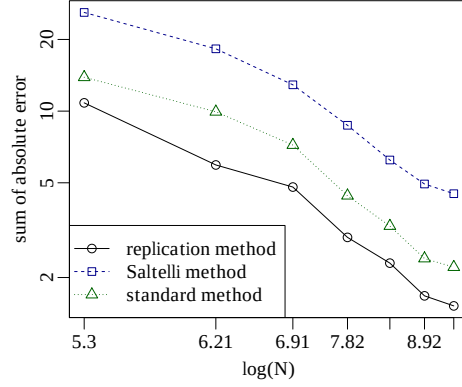
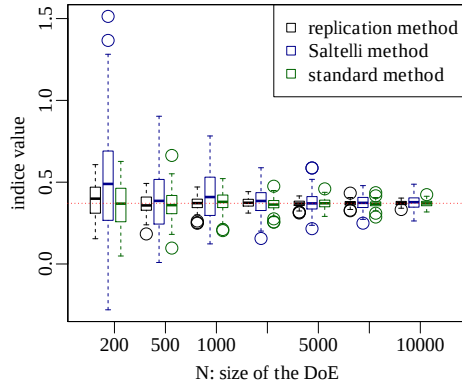
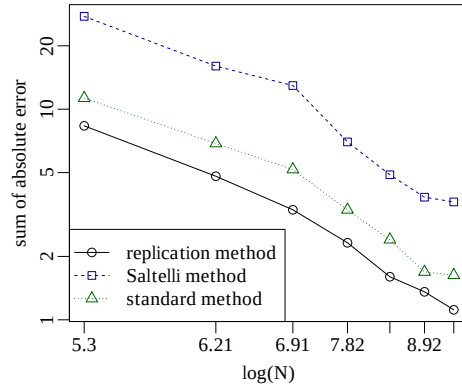
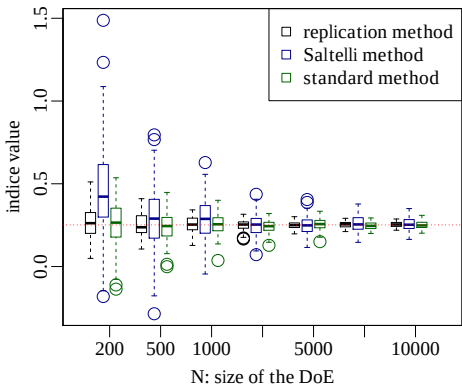
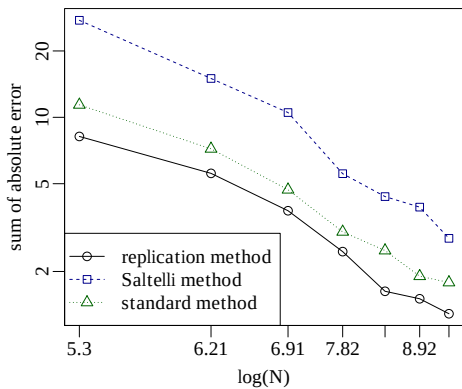
$$\sigma_1 = \sigma_2 = 0.3, \sigma_3 = 0.5, \sigma_4 = 0.6, \rho_{12} = 0.2$$

We use the same estimation procedure as in the last two subsections and we draw the results only for first-order Sobol' indices since the sum of the true values of those indices is equal to 0.999 which means that interactions are negligible. Looking at the results in Figure 3.10 our replication method still gives good results even in this case where there is no space filling strategy involved. The results given by the Saltelli method are in accordance with the results from the previous two subsections.

3.7 Conclusion

In this paper we proposed a methodology to estimate first-order or closed second-order grouped Sobol' indices using replicated designs to handle groups of dependent inputs. In our application on the LUTI model of Tranus this dependency translates into groups of linear ordered inputs. Thus, through an algorithm (Algorithm 3.2) we

FIGURE 3.10 – Estimation of first-order Sobol' indices given by the three methods for $r = 100$ repetitions and for different size N of the design. Left column : boxplot representation. Right column : log-log of the SAE_N .

(a) Sobol' index S_3 (b) Sobol' index S_3 (c) Sobol' index S_4 (d) Sobol' index S_4 (e) Sobol' index $S_{1,2}$ (f) Sobol' index $S_{1,2}$

presented a space filling strategy based on the simplex structure to sample points satisfying those constraints and a detailed procedure to estimate the associated Sobol' indices. This sampling method is comparable to a quasi-periodic distribution offering a good compromise between uniformity and regularity properties. Compared to the standard method of Sobol' [Sob93] or to Saltelli's approach [Sal02], our new estimation procedure has a better precision for a given cost, especially in the case of high number of inputs.

Acknowledgments

The authors would like to thank the anonymous reviewer and the associate editor for their helpful comments which substantially improved this paper. This work is supported by the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020).

3.A Illustration of the replication method for the estimation of first-order indices

$$D = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 \\ 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix}, \quad \Pi = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix}$$

Construction of D' :

$$\begin{aligned} \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} &\longrightarrow D = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix} \\ \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} &\longrightarrow D = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix} \\ \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} &\longrightarrow D = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix} \end{aligned}$$

Resulting in :

$$D' = \begin{pmatrix} 12 & 21 & 33 & 41 \\ 13 & 23 & 31 & 42 \\ 11 & 22 & 32 & 43 \end{pmatrix}$$

For a model f , the associated model outputs are :

$$Y_D = \begin{pmatrix} f(11, \textcolor{red}{21}, 31, 41) \\ f(\textcolor{blue}{12}, \textcolor{red}{22}, \textcolor{red}{32}, 42) \\ f(\textcolor{blue}{13}, \textcolor{red}{23}, \textcolor{red}{33}, 43) \end{pmatrix}, \quad Y_{D'} = \begin{pmatrix} f(\textcolor{blue}{12}, \textcolor{red}{21}, \textcolor{red}{33}, 41) \\ f(\textcolor{blue}{13}, \textcolor{red}{23}, 31, 42) \\ f(11, \textcolor{red}{22}, \textcolor{red}{32}, 43) \end{pmatrix}$$

To estimate the index S_1 we order Y_D with π_1 :

$$\pi_1 = \begin{pmatrix} 2 \\ 3 \\ 1 \end{pmatrix} \longrightarrow Y_{D_{new}} = \begin{pmatrix} f(\textcolor{blue}{12}, \textcolor{red}{22}, \textcolor{red}{32}, 42) \\ f(\textcolor{blue}{13}, \textcolor{red}{23}, \textcolor{red}{33}, 43) \\ f(11, \textcolor{red}{21}, 31, 41) \end{pmatrix}$$

Looking at both $Y_{D_{new}}$ and $Y_{D'}$, X_1 has been "frozen".

3.B Illustration of the replication method for the estimation of closed second-order indices

$$D = \begin{pmatrix} X_1 & X_2 & X_3 & X_4 \\ \textcolor{blue}{11} & \textcolor{red}{21} & 31 & 41 \\ \textcolor{blue}{12} & \textcolor{red}{22} & \textcolor{red}{32} & 42 \\ \textcolor{blue}{13} & \textcolor{red}{23} & \textcolor{red}{33} & 43 \end{pmatrix}, \quad \Pi = \begin{pmatrix} \pi_1 & \pi_2 & \pi_3 & \pi_4 \\ 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix}$$

Construction of a $2 - (3, 4, 1)$ orthogonal array $A = (M \oplus B)^T$:

$$A = \left(\left(\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix} \right) \right)^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 2 & 2 & 2 \\ 0 & 1 & 2 & 1 & 2 & 0 & 2 & 0 & 1 \\ 0 & 2 & 1 & 1 & 0 & 2 & 2 & 1 & 0 \end{pmatrix}^T$$

We can always add another column corresponding to the vector of elements of $GF(3)$ repeated 3 times. Then, elements of $GF(3)$ are substituted by $\{1, 2, 3\}$:

$$A = \begin{pmatrix} 0 & 0 & 0 & \textcolor{brown}{0} \\ 0 & 1 & 2 & \textcolor{brown}{1} \\ 0 & 2 & 1 & \textcolor{brown}{2} \\ 1 & 1 & 1 & \textcolor{brown}{0} \\ 1 & 2 & 0 & \textcolor{brown}{1} \\ 1 & 0 & 2 & \textcolor{brown}{2} \\ 2 & 2 & 2 & \textcolor{brown}{0} \\ 2 & 0 & 1 & \textcolor{brown}{1} \\ 2 & 1 & 0 & \textcolor{brown}{2} \end{pmatrix} \iff A = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix}$$

Construction of D_{OA} :

$$A = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow X = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix}$$

We iterate the procedure over the rows of A , resulting in :

$$D_{OA} = \begin{pmatrix} 13 & 23 & 33 & 43 \\ 13 & 21 & 32 & 41 \\ 13 & 22 & 31 & 42 \\ 11 & 21 & 31 & 43 \\ 11 & 22 & 33 & 41 \\ 11 & 23 & 32 & 42 \\ 12 & 22 & 32 & 43 \\ 12 & 23 & 31 & 41 \\ 12 & 21 & 33 & 42 \end{pmatrix}$$

Construction of D'_{OA} :

$$A = \begin{pmatrix} 3 & 3 & 3 & 3 \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix}$$

Then :

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ 1 & 2 & 2 & 3 \end{pmatrix} \longrightarrow D = \begin{pmatrix} 11 & 21 & 31 & 41 \\ 12 & 22 & 32 & 42 \\ 13 & 23 & 33 & 43 \end{pmatrix}$$

We iterate the procedure over the rows of A , resulting in :

$$D'_{OA} = \begin{pmatrix} 11 & 22 & 32 & 43 \\ 11 & 21 & 31 & 41 \\ 11 & 23 & 33 & 42 \\ 12 & 21 & 33 & 43 \\ 12 & 23 & 32 & 41 \\ 12 & 22 & 31 & 42 \\ 13 & 23 & 31 & 43 \\ 13 & 22 & 33 & 41 \\ 13 & 21 & 32 & 42 \end{pmatrix}$$

For a model f , the associated model outputs are :

$$Y_{D_{OA}} = \begin{pmatrix} f(13, 23, 33, 43) \\ f(13, 21, 32, 41) \\ f(13, 22, 31, 42) \\ f(11, 21, 31, 43) \\ f(11, 22, 33, 41) \\ f(11, 23, 32, 42) \\ f(12, 22, 32, 43) \\ f(12, 23, 31, 41) \\ f(12, 21, 33, 42) \end{pmatrix}, \quad Y_{D'_{OA}} = \begin{pmatrix} f(11, 22, 32, 43) \\ f(11, 21, 31, 41) \\ f(11, 23, 33, 42) \\ f(12, 21, 33, 43) \\ f(12, 23, 32, 41) \\ f(12, 22, 31, 42) \\ f(13, 23, 31, 43) \\ f(13, 22, 33, 41) \\ f(13, 21, 32, 42) \end{pmatrix}$$

To estimate the index $\underline{S}_{12}$ we order $Y_{D_{OA}}$ the following way :
denote by A_1 and A_2 the first two columns of A :

$$A_1 + A_2 * 3 - 3 = \begin{pmatrix} 3 \\ 3 \\ 3 \\ 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \end{pmatrix} + \begin{pmatrix} 3 \\ 1 \\ 2 \\ 1 \\ 2 \\ 3 \\ 2 \\ 3 \\ 1 \end{pmatrix} * 3 - 3 = \begin{pmatrix} 9 \\ 3 \\ 6 \\ 1 \\ 4 \\ 7 \\ 5 \\ 8 \\ 2 \end{pmatrix}$$

We order Y with this new permutation resulting in $Y_{D_{OA}}$:

$$\left\{ \begin{array}{l} Y[9] = Y_{D_{OA}}[1] = f(\mathbf{13}, \mathbf{23}, \mathbf{33}, 43) \\ Y[3] = Y_{D_{OA}}[2] = f(\mathbf{13}, \mathbf{21}, \mathbf{32}, 41) \\ Y[6] = Y_{D_{OA}}[3] = f(\mathbf{13}, \mathbf{22}, \mathbf{31}, 42) \\ Y[1] = Y_{D_{OA}}[4] = f(\mathbf{11}, \mathbf{21}, \mathbf{31}, 43) \\ Y[4] = Y_{D_{OA}}[5] = f(\mathbf{11}, \mathbf{22}, \mathbf{33}, 41) \\ Y[7] = Y_{D_{OA}}[6] = f(\mathbf{11}, \mathbf{23}, \mathbf{32}, 42) \\ Y[5] = Y_{D_{OA}}[7] = f(\mathbf{12}, \mathbf{22}, \mathbf{32}, 43) \\ Y[8] = Y_{D_{OA}}[8] = f(\mathbf{12}, \mathbf{23}, \mathbf{31}, 41) \\ Y[2] = Y_{D_{OA}}[9] = f(\mathbf{12}, \mathbf{21}, \mathbf{33}, 42) \end{array} \right.$$

Next, we order $Y_{D'_{OA}}$ the following way :
denote by A_π the matrix obtained as follows :

$$A = \begin{pmatrix} \boxed{3} & \boxed{3} & \boxed{3} & \boxed{3} \\ 3 & 1 & 2 & 1 \\ 3 & 2 & 1 & 2 \\ 1 & 1 & 1 & 3 \\ 1 & 2 & 3 & 1 \\ 1 & 3 & 2 & 2 \\ 2 & 2 & 2 & 3 \\ 2 & 3 & 1 & 1 \\ 2 & 1 & 3 & 2 \end{pmatrix} \longrightarrow \Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix}$$

and :

$$\Pi = \begin{pmatrix} 2 & 1 & 3 & 1 \\ 3 & 3 & 1 & 2 \\ \boxed{1} & \boxed{2} & \boxed{2} & \boxed{3} \end{pmatrix} \longrightarrow A_\pi = \begin{pmatrix} 1 & 2 & 2 & 3 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

denote by A_{π_1} and A_{π_2} the first two columns of A_π :

$$A_{\pi_1} + A_{\pi_2} * 3 - 3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 2 \\ 2 \\ 2 \\ 3 \\ 3 \\ 3 \end{pmatrix} + \begin{pmatrix} 2 \\ 1 \\ 3 \\ 1 \\ 3 \\ 2 \\ 3 \\ 2 \\ 1 \end{pmatrix} * 3 - 3 = \begin{pmatrix} 4 \\ 1 \\ 7 \\ 2 \\ 8 \\ 5 \\ 9 \\ 6 \\ 3 \end{pmatrix}$$

we order Y with this new permutation resulting in $Y_{D'_{OA}}$:

$$\left\{ \begin{array}{l} Y'[4] = Y_{D'_{OA}}[1] = f(11, \textcolor{red}{22}, \textcolor{red}{32}, 43) \\ Y'[1] = Y_{D'_{OA}}[2] = f(11, \textcolor{red}{21}, 31, 41) \\ Y'[7] = Y_{D'_{OA}}[3] = f(11, \textcolor{red}{23}, \textcolor{red}{33}, 42) \\ Y'[2] = Y_{D'_{OA}}[4] = f(\textcolor{blue}{12}, \textcolor{red}{21}, \textcolor{red}{33}, 43) \\ Y'[8] = Y_{D'_{OA}}[5] = f(\textcolor{blue}{12}, \textcolor{red}{23}, \textcolor{red}{32}, 41) \\ Y'[5] = Y_{D'_{OA}}[6] = f(\textcolor{blue}{12}, \textcolor{red}{22}, 31, 42) \\ Y'[9] = Y_{D'_{OA}}[7] = f(\textcolor{blue}{13}, \textcolor{red}{23}, 31, 43) \\ Y'[6] = Y_{D'_{OA}}[8] = f(\textcolor{blue}{13}, \textcolor{red}{22}, \textcolor{red}{33}, 41) \\ Y'[3] = Y_{D'_{OA}}[9] = f(\textcolor{blue}{13}, \textcolor{red}{21}, \textcolor{red}{32}, 42) \end{array} \right.$$

Looking at both Y and Y' , the two inputs X_1, X_2 have been "frozen" :

$$Y = \begin{pmatrix} f(11, \textcolor{red}{21}, 31, 43) \\ f(\textcolor{blue}{12}, \textcolor{red}{21}, \textcolor{red}{33}, 42) \\ f(\textcolor{blue}{13}, \textcolor{red}{21}, \textcolor{red}{32}, 41) \\ f(11, \textcolor{red}{22}, \textcolor{red}{33}, 41) \\ f(\textcolor{blue}{12}, \textcolor{red}{22}, \textcolor{red}{32}, 43) \\ f(\textcolor{blue}{13}, \textcolor{red}{22}, 31, 42) \\ f(11, \textcolor{red}{23}, \textcolor{red}{32}, 42) \\ f(\textcolor{blue}{12}, \textcolor{red}{23}, 31, 41) \\ f(\textcolor{blue}{13}, \textcolor{red}{23}, \textcolor{red}{33}, 43) \end{pmatrix} \quad Y' = \begin{pmatrix} f(11, \textcolor{red}{21}, 31, 41) \\ f(\textcolor{blue}{12}, \textcolor{red}{21}, \textcolor{red}{33}, 43) \\ f(\textcolor{blue}{13}, \textcolor{red}{21}, \textcolor{red}{32}, 42) \\ f(11, \textcolor{red}{22}, \textcolor{red}{32}, 43) \\ f(\textcolor{blue}{12}, \textcolor{red}{22}, 31, 42) \\ f(\textcolor{blue}{13}, \textcolor{red}{22}, \textcolor{red}{33}, 41) \\ f(11, \textcolor{red}{23}, \textcolor{red}{33}, 42) \\ f(\textcolor{blue}{12}, \textcolor{red}{23}, \textcolor{red}{32}, 41) \\ f(\textcolor{blue}{13}, \textcolor{red}{23}, 31, 43) \end{pmatrix}$$

3.C Illustration of Algorithm 2 for k=2 and n=9

Creation of the 4 vertices of the unit 2-hypercube and numbering :

$$\begin{array}{c} X_1 \quad X_2 \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \end{array}$$

Identification of the vertices satisfying the constraint $X_1 \leq X_2$ and creation of the container matrix M_{coord} :

$$\left\{ \begin{array}{l} 0 \leq 0 \\ 1 \not\leq 0 \\ 0 \leq 1 \\ 1 \leq 1 \end{array} \right. \longrightarrow M_{coord} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

Construction of the $2!$ vectors of permutations and their corresponding vertices matrices (obtained by column-wise permutations of M_{coord}) :

$$v_1 = \begin{pmatrix} 1 & 2 \end{pmatrix} \longrightarrow M_1 = M_{coord} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}$$

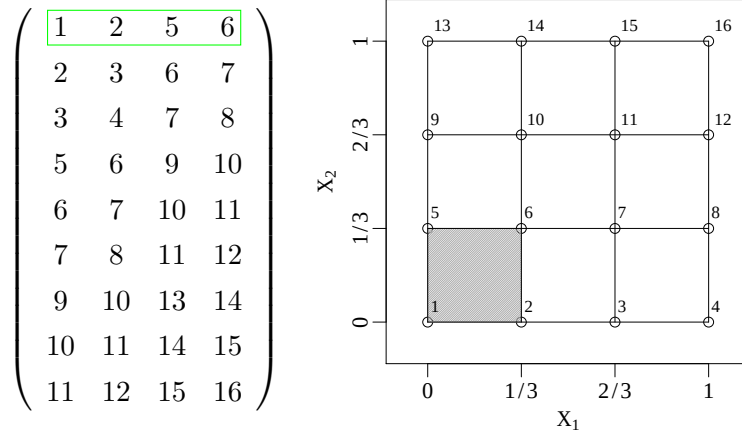
$$v_2 = \begin{pmatrix} 2 & 1 \end{pmatrix} \longrightarrow M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

Retrieve the index vectors w_j of matching vertices between each M_j and the initial matrix of vertices :

$$M_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow \begin{matrix} \boxed{1} \\ 2 \\ \boxed{3} \\ \boxed{4} \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow w_1 = \begin{pmatrix} 1 & 3 & 4 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \longrightarrow \begin{matrix} \boxed{1} \\ \boxed{2} \\ 3 \\ \boxed{4} \end{matrix} \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \longrightarrow w_2 = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix}$$

The matrix of subdivision of the 2-hypercube into $3^2 = 9$ sub-hypercubes (one sub-hypercube per row of the matrix) writes as follows :



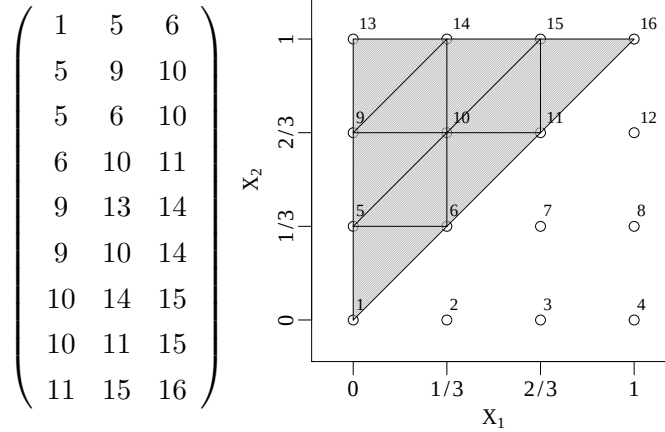
where each sub-hypercube is identified by its vertices numerotations. For each sub-hypercube and for each w_j conserve the vertices whose index matches w_j :

$$\left\{ \begin{array}{l} w_1 = \begin{pmatrix} 1 & 3 & 4 \end{pmatrix} \longrightarrow \begin{pmatrix} \boxed{1} & 2 & \boxed{5} & \boxed{6} \end{pmatrix} \\ w_2 = \begin{pmatrix} 1 & 2 & 4 \end{pmatrix} \longrightarrow \begin{pmatrix} \boxed{1} & \boxed{2} & 5 & \boxed{6} \end{pmatrix} \end{array} \right.$$

Iterate the process over the sub-hypercube and conserve only the vertices satisfying the constraint $X_1 \leq X_2$:

$$\left\{ \begin{array}{l} w_1 = \begin{pmatrix} 1 & 5 & 6 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \leq 0 \\ 0 \leq \frac{1}{3} \\ \frac{1}{3} \leq \frac{1}{3} \end{pmatrix} \longrightarrow \text{accept} \\ w_2 = \begin{pmatrix} 1 & 2 & 6 \end{pmatrix} \longrightarrow \begin{pmatrix} 0 \leq 0 \\ \frac{1}{3} \not\leq 0 \\ \frac{1}{3} \leq \frac{1}{3} \end{pmatrix} \longrightarrow \text{reject} \end{array} \right.$$

This gives a total of $3^2 = 9$ simplices verifying the constraints. The vertices of the 9 simplices are given by the rows of the following matrix :



We then sample one point in each of the 9 simplices with Theorem 3.2.

Recursive estimation procedure of Sobol' indices based on replicated designs

Les résultats de ce chapitre ont donné lieu à un article soumis pour publication [GAMP16]. L'objectif de ce chapitre est d'apporter une réponse à la problématique P_3 soulevée en introduction portant sur l'estimation récursive des indices de Sobol'. Nous proposons dans ce chapitre une version récursive de la méthode répliquée pour estimer les indices de Sobol' d'ordre un ou d'ordre deux fermés. L'idée d'une approche récursive est d'estimer de plus en plus précisément les indices de Sobol', en conservant à chaque étape les évaluations déjà effectuées pour l'estimation des indices à l'étape précédente. Le point clé est donc l'utilisation de plans emboîtés. Le nombre d'itérations pour l'estimation récursive est réglé par un critère d'arrêt tel qu'un seuil relatif à la valeur absolue de la différence entre deux estimations consécutives. La méthode de construction du plan emboîté est spécifique à chaque cas (indice d'ordre un et indice fermé d'ordre deux).

Pour l'estimation des indices d'ordre un, nous exploitons la structure d'hypercube latin emboîté introduite par Qian [Qia09]. Ce plan correspond à un hypercube latin contenant plusieurs blocs où chaque bloc possède lui-même une structure d'hypercube latin en projection.

Pour l'estimation des indices fermés d'ordre deux, nous proposons deux méthodes pour construire un tableau orthogonal de force deux emboîté. La première est une procédure d'acceptation-rejet. La seconde est une méthode dite algébrique s'appuyant sur des résultats de théorie du codage.

Nous proposons d'étudier les propriétés d'uniformité et de remplissage de l'espace des plans emboîtés construits. Par ailleurs, nous comparons à l'aide d'une fonction test la méthode répliquée récursive par rapport à sa version d'origine non récursive.

Titre : Recursive estimation procedure of Sobol' indices based on replicated designs.

Auteurs : Laurent Gilquin^{a,b}, Elise Arnaud^b, Hervé Monod^c, Clémentine Prieur^b.

a Inria Grenoble - Rhône-Alpes, Inovallée, 655 avenue de l'Europe, 38330 Montbonnot

b Univ. Grenoble Alpes, Jean Kuntzmann Laboratory, F-38000 Grenoble, France

CNRS, LJK, F-38000 Grenoble, France, Inria

c MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-En-Josas, France

4.1 Introduction

Mathematical models used in various fields are often quite complex. The behavior of some of these models may only be explored through the study of uncertainties propagated from their inputs. Sensitivity analysis studies how the uncertainty on an output of a mathematical model can be attributed to sources of uncertainty among the inputs. Among the large number of available approaches, the variance-based method introduced by Sobol' [Sob93] relies on the calculation of sensitivity measures called Sobol' indices. The method is based on a variance decomposition of the model output into fractions which can be attributed to sets of inputs, assuming that the uncertainty on the sets of inputs is modeled by independent probability distributions. The influences of each set are summarized by the Sobol' indices which are scalars between 0 and 1. The higher the index the more influential the set. One can distinguish first-order indices that estimate the main effect of each set of inputs from higher-order indices that estimate the corresponding order of interactions between sets of inputs. Various procedures have been proposed in the literature (see Saltelli [SCS08] for a survey) to estimate Sobol' indices. Unfortunately, these procedures require a significant number of model evaluations that can be prohibitive for expansive models. A solution reducing this number lies in the use of replicated designs.

The notion of replicated designs was introduced by McKay [McK95]. Later on, Mara *et al.* [MJ08] combine these designs with "pick-freeze" estimators [Sob93] to estimate first-order Sobol' indices. This procedure, called replication procedure, has been further studied and generalized in Tissot *et al.* [TP15] to the estimation of closed second-order indices. This generalization relies on the construction of orthogonal arrays (OA) (see [HSS99]). The replication procedure has the major advantage of reducing considerably the estimation cost as it requires to construct only two replicated designs of size n . However, if the input space is not properly explored, that is if n is too small, the Sobol' indices estimates may not be accurate enough.

To address this challenge, we need a procedure to sequentially add new points to an initial design and a recursive formula of the Sobol' index estimator. Adding

new points is straightforward when the initial design is composed with independent and identically distributed points. However, in the replication procedure as introduced in [TP15], the initial design possesses either a structure of Latin hypercube or orthogonal array whether first- or closed second-order Sobol' indices are estimated. To preserve these structures, we focus on the construction of nested space-filling designs. An algorithm for the construction of nested Latin hypercubes has been proposed by Qian [Qia09]. It allows to double the size of the design at each step. Our approach to render the replication procedure recursive for the estimation of first-order Sobol' indices is based on this construction.

The main contributions of this paper are two constructions of a nested orthogonal array for the recursive estimation of closed second-order indices with the replication procedure. Each construction starts with an initial orthogonal array and updates it sequentially by adding a fixed number of new points. Constructions of nested orthogonal arrays have already been studied in [QTW09], [QAW09] and [Dey12]. These latter suffer from at least one of the following drawbacks :

- The size of the initial design is rather large, hence at each step a large number of new points is added.
- The constructions deal only with specific values of the input space dimension.
- The discretization is not the same in each dimension, more precisely only one dimension is finely discretized.

Conversely, the two constructions we propose do not suffer from these drawbacks. The paper is organized as follows. In Section 4.2, backgrounds are given on Sobol' indices and the replication procedure. Then, the process rendering the replication procedure recursive is detailed. Section 4.3 deals with the construction of the nested space-filling designs : nested Latin hypercube and nested orthogonal arrays. In Section 4.4, regularity and uniformity properties of these two designs are studied. The end of this section is devoted to an application of our recursive procedure to a toy example.

4.2 Recursive estimation of Sobol' indices

4.2.1 Definition of Sobol' indices

Consider the following model defined from a black box perspective :

$$f: \begin{cases} \mathbb{R}^d & \rightarrow \mathbb{R} \\ \mathbf{x} = (x_1, \dots, x_d) & \mapsto y = f(\mathbf{x}) \end{cases} \quad (4.1)$$

where y is the output of the model f , $\mathbf{x}$ the input vector and d the dimension of the input space. Denote by $\subsetneq$ the proper (strict) inclusion symbol and by $\subseteq$ the inclusion symbol.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. We model the uncertainty on the inputs by a random vector $\mathbf{X} = (X_1, \dots, X_d)$ whose components are independent. Let

$u \subseteq \mathcal{D}$, $\mathbf{X}_u$ denotes a vector with components X_j , $j \in u$. Let $P_{\mathbf{X}} = P_{X_1} \otimes \dots \otimes P_{X_d}$ be the distribution of $\mathbf{X}$. We assume that $f \in \mathbb{L}^2(P_{\mathbf{X}})$. The model f can then be uniquely decomposed into summands of increasing dimensions (functional ANOVA decomposition [Sob93, Hoe48]) :

$$f(\mathbf{X}) = f_0 + \sum_j f_j(X_j) + \sum_{k < l} f_{k,l}(X_k, X_l) + \dots + f_{1,\dots,d}(X_1, \dots, X_d) , \quad (4.2)$$

where $E[f_u(\mathbf{X}_u)f_w(\mathbf{X}_w)] = 0$, $\forall (u, w) \subseteq \{1, \dots, d\}^2$, $u \neq w$. Denote $Y = f(\mathbf{X})$, this implies that $f_0 = E[Y]$ and that the components are mutually orthogonal with respect to $P_{\mathbf{X}}$. Let $u \subseteq \{1, \dots, d\}$, each component is defined by :

$$f_u(\mathbf{X}_u) = E[Y|\mathbf{X}_u] - \sum_{v \subsetneq u} f_v(\mathbf{X}_v) .$$

The functional decomposition can be used to measure the global sensitivity of the output Y to $\mathbf{X}_u$. By squaring and integrating (4.2), due to orthogonality we get :

$$\sigma^2 = \text{Var}[Y] = \sum_j \sigma_j^2 + \sum_{k < l} \sigma_{k,l}^2 + \dots + \sigma_{1,\dots,d}^2 , \quad (4.3)$$

where :

$$\sigma_u^2 = \text{Var}[f_u(\mathbf{X}_u)] = \text{Var}[E[Y|\mathbf{X}_u]] - \sum_{v \subsetneq u} \sigma_v^2 .$$

Resulting from this decomposition, the Sobol' indices are defined by :

$$S_u = \frac{\sigma_u^2}{\sigma^2} .$$

Let $|u|$ denote the cardinal of u . The Sobol' index S_u measures the contribution to σ^2 of the $|u|^{\text{th}}$ -order interaction between the X_j , $j \in u$. Closed Sobol' indices are defined by :

$$\underline{S}_u = \frac{\text{Var}[E[Y|\mathbf{X}_u]]}{\sigma^2} .$$

The closed Sobol' index $\underline{S}_u$ measures the contribution of the X_j , $j \in u$, by themselves or in interaction with each other. As an example, if $u = \{k, l\}$, $k \neq l$, then $\underline{S}_{k,l} = S_{k,l} + S_k + S_l$. At last, note that :

$$\sum_{u \subseteq \{1, \dots, d\}, u \neq \emptyset} S_u = 1 ,$$

which gives a direct interpretation of the value of each index. Most of the time, no explicit formulation of Sobol' indices is available. To bypass this problem, one needs to resort to estimation methods.

4.2.2 Estimation of Sobol' indices

In this section, we review succinctly the estimation procedure introduced by Sobol' [Sob93]. Consider $\mathbf{X}$ and $\mathbf{X}'$ two independent vectors distributed as the input vector. Let $u \subseteq \mathcal{D}$ and denote by $-u$ its complement. The hybrid point $\mathbf{W} = (\mathbf{X}_u : \mathbf{X}'_{-u})$ is defined by $W_j = X_j$ if $j \in u$ and $W_j = X'_j$ otherwise. We define the following model outputs : $Y = f(\mathbf{X})$, $Y_u = f(\mathbf{X}_u : \mathbf{X}'_{-u})$.

To estimate $\underline{S}_u$, we start from the formula introduced by Janon *et al.* [JKL⁺14, Lemma 1.2] that expresses the Sobol' index as a regression coefficient between the two outputs Y and Y_u :

$$\underline{S}_u = \frac{\text{Cov}(Y, Y_u)}{\text{Var}[Y]} .$$

Then, we proceed as in [Sob93] and introduce two designs, each of size n :

$$\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n, \quad \mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n .$$

$\mathcal{P}$ (resp. $\mathcal{P}'$) is a matrix where each row is a point $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,d})$ (resp. $\mathbf{X}'_i$) of the input space and each column contains n realizations $X_{i,j}$ of each input X_j , $j = 1, \dots, d$. A third design $\mathcal{P}^u = \{\mathbf{X}_{i,u} : \mathbf{X}'_{i,-u}\}_{i=1}^n$ is constructed from $\mathcal{P}$ and $\mathcal{P}'$ by columns substitution. By evaluating the model with $\mathcal{P}$ and $\mathcal{P}^u$, we obtain n realizations of Y and Y_u noted $\{Y_i\}_{i=1}^n$ and $\{Y_{i,u}\}_{i=1}^n$. Following [MNM06], we consider the estimator :

$$\hat{\underline{S}}_u = \frac{\frac{1}{n} \sum_{i=1}^n Y_i Y_{i,u} - \left(\frac{1}{n} \sum_{i=1}^n Y_i \right) \left(\frac{1}{n} \sum_{i=1}^n Y_{i,u} \right)}{\frac{1}{n} \sum_{i=1}^n (Y_i)^2 - \left(\frac{1}{n} \sum_{i=1}^n Y_i \right)^2} . \quad (4.4)$$

Other choices are possible for the estimator (see [Owe13a] for a succinct review). We focus on (4.4) whose asymptotic properties have been studied in [JKL⁺14].

The main drawback of the aforementioned procedure is the high number of model evaluations required. Estimating all first-order (resp. all closed second-order) Sobol' indices costs $n(d+1)$ (resp. $n(\binom{d}{2} + 1)$) model evaluations. The larger n , the more accurate the estimation of Sobol' indices. Some improvements have been introduced by Saltelli [Sal02] to reduce the number of evaluations but with a cost still depending on the input space dimension. A solution reducing drastically this costs lies in the use of replicated designs.

4.2.3 Replication procedure and associated designs

In the following, we review the procedure based on replicated designs to estimate first- or closed second-order Sobol' indices. We refer to it as replication procedure. We assume that the inputs $X_1, \dots, X_d$ are independent and uniformly distributed on $[0, 1]$. The generalization to other product distributions is provided in Remark 4.1 page 100.

The concept of replicated designs was first introduced by McKay in [McK95]. Here, we define it as follows :

Definition 4.1. Let $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ be two designs in $[0, 1]^d$. Let $\mathcal{P}^u = \{\mathbf{X}_{i,u}\}_{i=1}^n$ (resp. $\mathcal{P}'^u$), $u \subsetneq \mathcal{D}$, denote the subset of dimensions (columns) of $\mathcal{P}$ (resp. $\mathcal{P}'$) indexed by u . We say that $\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order $a \in \{1, \dots, d-1\}$ if for any $u \subsetneq \mathcal{D}$ such that $|u| = a$, $\mathcal{P}^u$ and $\mathcal{P}'^u$ are the same point set in $[0, 1]^a$. We define by π_u the permutation that rearranges the rows of $\mathcal{P}'^u$ into $\mathcal{P}^u$.

Example Consider the two designs :

$$\mathcal{P} = \begin{pmatrix} 0.08 & 0.46 & 0.21 \\ 0.15 & 0.77 & 0.43 \\ 0.89 & 0.30 & 0.05 \\ 0.70 & 0.23 & 0.95 \end{pmatrix}, \quad \mathcal{P}' = \begin{pmatrix} 0.89 & 0.30 & 0.95 \\ 0.15 & 0.23 & 0.21 \\ 0.70 & 0.46 & 0.43 \\ 0.08 & 0.77 & 0.05 \end{pmatrix}.$$

$\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order 1. $\forall i$, the i -th columns of $\mathcal{P}$ and $\mathcal{P}'$ share the same unordered set of values. The permutation $\pi_1 = (4, 2, 1, 3)$ order the first column of $\mathcal{P}'$ into the first column of $\mathcal{P}$.

The key point of the replication procedure is to construct two replicated designs and use their structure to mimic the hybrid points $\{\mathbf{X}_{i,u} : \mathbf{X}'_{i,-u}\}_{i=1}^n$. More precisely, let $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ be two replicated designs of order $|u|$. Denote by $\{Y_i\}_{i=1}^n$ and $\{Y'_i\}_{i=1}^n$ the two sets of model evaluations obtained with $\mathcal{P}$ and $\mathcal{P}'$. From Definition 4.1, we know that $\mathbf{X}'_{\pi_u(i),u} = \mathbf{X}_{i,u}$.

$$\begin{aligned} \text{Then, } Y'_{\pi_u(i)} &= f(\mathbf{X}'_{\pi_u(i),u} : \mathbf{X}'_{\pi_u(i),-u}) , \\ &= f(\mathbf{X}_{i,u} : \mathbf{X}'_{\pi_u(i),-u}) . \end{aligned}$$

Hence, each Sobol' index $\underline{S}_u$ can be estimated via formula (4.4) with $Y'_{\pi_u(i)}$ in place of $Y_{i,u}$ without requiring further model evaluations. As such, the cost of the procedure equals $2n$ to estimate either all first-order or all closed second-order indices. We detail below the structure of the two replicated designs for each case.

Estimation of first-order indices There are various choices for constructing two replicated designs of order 1. In [MJ08], $\mathcal{P}$ and $\mathcal{P}'$ are composed with i.i.d points. In [TP15], the authors propose to use Latin hypercube designs insuring most of the time a better exploration of the input space :

Definition 4.2 (Latin hypercube design). Denote by Π_n the set of all the permutations of $\{1, \dots, n\}$ and let $\pi_1, \dots, \pi_d$ be d independent random variables uniformly distributed on Π_n . $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ is a Latin hypercube design if :

$$\mathbf{X}_i = \left(\frac{\pi_1(i) - U_{i,1}}{n}, \dots, \frac{\pi_d(i) - U_{i,d}}{n} \right), \quad (4.5)$$

where the $U_{i,j}$ are independent random variables uniformly distributed on $[0, 1]$ and independent of the π_j .

The first replicated design $\mathcal{P}$ is constructed with Definition 4.2 above, then $\mathcal{P}'$ is obtained by permuting independently the values of each column of $\mathcal{P}$. In [TP15], the two resulting designs are referred as replicated Latin hypercube designs.

Estimation of closed second-order indices The generalization to second-order indices was introduced in [TP15]. To estimate these indices, one needs to find a structure that “freezes” each subset of two variables. A solution relies on the use of orthogonal arrays [HSS99, Definition 1.1] :

Definition 4.3 (Orthogonal array). *A $n \times d$ array $A = \{\mathbf{A}_i\}_{i=1}^n$, $\mathbf{A}_i = (A_{i,1}, \dots, A_{i,d})$, with values from a set S of cardinality q is said to be an orthogonal array with q levels, strength t ($0 \leq t \leq d$) and index λ if every $n \times t$ sub-array of A contains each t -tuple based on S exactly λ times as a row. The orthogonal array A satisfies $n = \lambda q^t$. It is denoted by $OA_\lambda(q, d, t)$.*

Here, the space S is identified as the Galois field of order q , denoted by $GF(q)$, where q is a prime number or prime power number ($q = p^\alpha$, p prime and $\alpha \in \mathbb{N}$). Once an orthogonal array is constructed its levels are substituted by $1, \dots, q$, q indicating the number of points into which each input is discretized. For constructions of orthogonal arrays we invite the reader to consult [HSS99].

From Definition 4.3, orthogonal arrays of strength two are replicated designs of order 2 and as such naturally “freeze” each subset of two variables. Therefore, the strategy proposed in [TP15] is to construct two replicated orthogonal arrays of strength two :

Definition 4.4 (Replicated orthogonal arrays). *Let $A = \{\mathbf{A}_i\}_{i=1}^{q^t}$ be an $OA_1(q, d, t)$. Denote by Π_q the set of all the permutations of $\{1, \dots, q\}$ and let $\pi_1, \dots, \pi_d$ be d independent random variables uniformly distributed on Π_q . $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^{q^t}$ and $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^{q^t}$ are two replicated orthogonal arrays if :*

$$\begin{aligned} \mathbf{X}_i &= \left(\frac{A_{i,1} - U_{i,1}}{q}, \dots, \frac{A_{i,d} - U_{i,d}}{q} \right), \\ \mathbf{X}'_i &= \left(\frac{\pi_1(A_{i,1}) - U_{\pi_1(A_{i,1}),1}}{q}, \dots, \frac{\pi_d(A_{i,d}) - U_{\pi_d(A_{i,d}),d}}{q} \right), \end{aligned} \quad (4.6)$$

where the $U_{i,j}$ are independent random variables uniformly distributed on $[0, 1]$ and independent of the π_j .

Note that this definition is given in a Monte-Carlo context (the points lies in $[0, 1]^d$). The construction of $\mathcal{P}'$ reduces to apply independently a permutation to

the symbols of each column of A . We note by $\diamond$ the operator achieving this rearrangement :

$$A' = \diamond(A, \{\pi_1, \dots, \pi_d\}) \Leftrightarrow \mathbf{A}'_i = (\pi_1(A_{i,1}), \dots, \pi_d(A_{i,d})), \quad i = 1, \dots, q^t. \quad (4.7)$$

Using two replicated orthogonal arrays of strength two, the cost of the replication procedure writes $2n$ where $n = q^2$.

Remark 4.1. *In this section, the constructions of designs $\mathcal{P}$ and $\mathcal{P}'$ are only valid when dealing with variables $X_1, \dots, X_d$ independent and uniformly distributed on $[0, 1]$. However, these constructions can be generalized to other non-uniform distributions. Denote by $F_1, \dots, F_d$ the cumulative distribution functions of $X_1, \dots, X_d$. In the general case, the two designs $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ are constructed as follows :*

$$\begin{aligned} \mathbf{X}_i &= (F_1^{-1}(X_{i,1}), \dots, F_d^{-1}(X_{i,d})), \\ \mathbf{X}'_i &= (F_1^{-1}(X'_{i,1}), \dots, F_d^{-1}(X'_{i,d})), \end{aligned}$$

where $F_1^{-1}, \dots, F_d^{-1}$ are the quantile functions of $X_1, \dots, X_d$.

4.2.4 Recursive procedure

To estimate Sobol' indices with one of the latter estimation methods, one needs to fix a value for the size n of the designs. In practice, one wants to choose n large enough to ensure a sufficient precision on the Sobol' estimates while keeping an affordable computational time. This choice is difficult to address mostly because it depends on the complexity of the model studied. As such, it is hard to bring out a general rule of thumb.

The practical solution investigated here is to increase the accuracy of the estimates by sequentially adding new points. To do so, we propose a recursive version of the replication procedure. First, the construction of the two replicated designs is carried out according to the following scheme :

$$\begin{cases} \mathcal{P}_0 = B_0 \\ \mathcal{P}_\ell = \mathcal{P}_{\ell-1} \cup B_\ell \end{cases}, \quad \begin{cases} \mathcal{P}'_0 = B'_0 \\ \mathcal{P}'_\ell = \mathcal{P}'_{\ell-1} \cup B'_\ell \end{cases}, \quad \ell \geq 1,$$

where B_ℓ, B'_ℓ are the new sets of m_ℓ points added at step ℓ . In the following, we refer to these sets as blocks. The blocks B_ℓ and B'_ℓ are then used to evaluate the model and obtain two sets of outputs $\{Y^i\}_{i=n_{\ell-1}+1}^{n_\ell}$ and $\{Y_u^i\}_{i=n_{\ell-1}+1}^{n_\ell}$ where n_ℓ denote the size of designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$. Therefore, these two designs are nested designs partitioned into blocks.

Our recursive procedure requires to write down a recursive formula for the Sobol' index estimator. Recall the expression of the Sobol' index :

$$\underline{S}_u = \frac{\text{Cov}[Y, Y_u]}{\text{Var}[Y]} = \frac{\text{E}[Y Y_u] - \text{E}[Y] \text{E}[Y_u]}{\text{Var}[Y]}, \quad (4.8)$$

where Y and Y_u are evaluated with the two replicated designs. At step $\ell \geq 1$, the Sobol' index $\underline{S}_u$ is estimated by the following family of recursive estimators :

$$\hat{\underline{S}}_u^{(\ell)} = \frac{\phi_\ell - \psi_\ell \xi_\ell}{V_\ell}, \quad (4.9)$$

where ϕ_ℓ , ψ_ℓ , ξ_ℓ and V_ℓ are identified with formula (4.8) and are estimated at step $\ell = 0$ directly from blocks B_0 and B'_0 . These terms are recursively defined as follows :

$$\left\{ \begin{array}{l} n_\ell = n_{\ell-1} + m_\ell, \\ n_\ell \phi_\ell = n_{\ell-1} \phi_{\ell-1} + m_\ell \sum_{i=n_{\ell-1}+1}^{n_\ell} Y^i Y_u^i, \\ n_\ell \psi_\ell = n_{\ell-1} \psi_{\ell-1} + m_\ell \sum_{i=n_{\ell-1}+1}^{n_\ell} Y^i, \\ n_\ell \xi_\ell = n_{\ell-1} \xi_{\ell-1} + m_\ell \sum_{i=n_{\ell-1}+1}^{n_\ell} Y_u^i, \\ (n_\ell - 1)V_\ell = (n_{\ell-1} - 2)V_{\ell-1} + n_{\ell-1} \psi_{\ell-1}^2 + \sum_{i=n_{\ell-1}+1}^{n_\ell} (Y^i)^2 - n_\ell \psi_\ell^2. \end{array} \right.$$

Algorithm 4.1 summarizes the main steps of our recursive procedure. The form of the stopping criterion (variable **test**) is discussed in Section 4.4.2. The set $\mathcal{D}$ equals either $\{1, \dots, d\}$ or $\{(k, l) \in \{1, \dots, d\}^2; k < l\}$ whether first-order or closed second-order indices are estimated. The cost of our recursive procedure equals $2 \times \sum_{\ell \geq 0} m_\ell$.

Algorithm 4.1 Recursive estimation of Sobol' indices

- 1: $\ell \leftarrow 0$, $\hat{\underline{S}}_u^{(0)} \leftarrow 0$, **test** $\leftarrow$ true
 - 2: $\mathcal{P}_0 \leftarrow B_0$
 $\mathcal{P}'_0 \leftarrow B'_0$
 - 3: **while** **test** **do**
 - 4: **for** $u \in \mathcal{D}$ **do**
 - 5: Compute Y and Y_u from B_ℓ and B'_ℓ
 - 6: Evaluate $\hat{\underline{S}}_u^{(\ell)}$ with (4.9)
 - 7: **test** $\leftarrow$ stopping criterion
 - 8: $\mathcal{P}_{\ell+1} \leftarrow \mathcal{P}_\ell \cup B_{\ell+1}$
 $\mathcal{P}'_{\ell+1} \leftarrow \mathcal{P}'_\ell \cup B'_{\ell+1}$
 - 9: $\ell \leftarrow \ell + 1$
 - 10: Return the Sobol' estimates
-

In the next section, we detail the construction of the two nested designs required for the estimation of either first-order or closed second-order Sobol' indices. In both cases, the construction ensures that at each step $\ell \geq 0$, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ possess a space-filling structure and are two replicated designs of order 1 or 2.

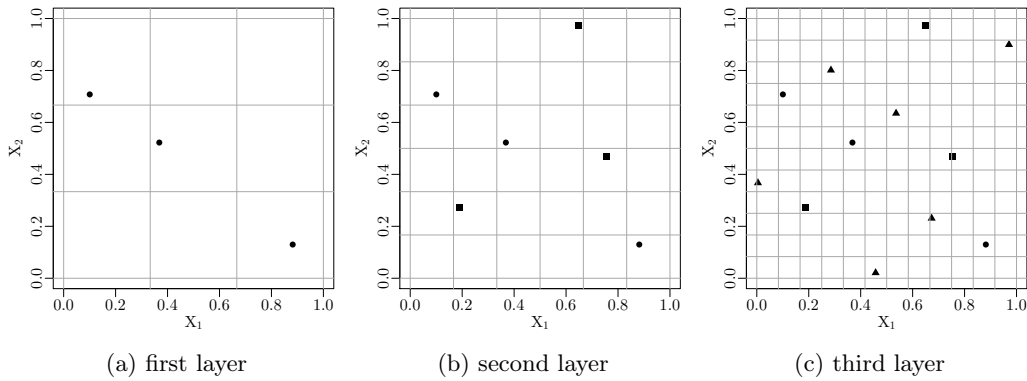
4.3 Nested space-filling designs

For the estimation of first-order indices, the two nested designs are nested Latin hypercube designs. The number of blocks partitioning the design has to be specified beforehand. This number defines the initial discretization of the input space. This discretization is then further refined with the addition of a new block. For the estimation of closed second-order indices, the two nested designs are nested orthogonal arrays of strength two. The number of blocks is iteratively augmented. However, the discretization of each input is fixed at the first step and remains unvaried throughout the procedure.

4.3.1 Nested Latin hypercube design

A way to augment the number of points while preserving a Latin hypercube structure has been proposed by Qian in [Qia09]. A nested Latin hypercube design is a Latin hypercube design partitioned into blocks defining multiple layers. As an illustration, a two dimensional nested Latin hypercube design with 3 layers is presented in Figure 4.1. Each layer possesses itself a Latin hypercube structure in a grid progressively refined. The algorithm underlying the construction of a nested Latin hypercube design is detailed in [Qia09, Section 5].

FIGURE 4.1 – Nested Latin hypercube design with three layers (a), (b), (c). The symbols mark the new points (i.e the blocks) added at each step (in order circle, square, triangle).



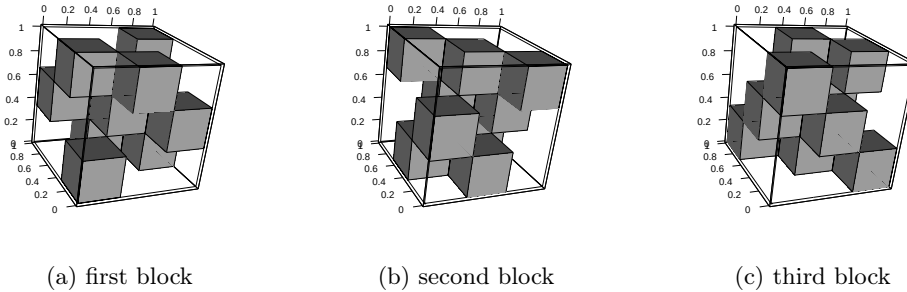
For the estimation of first-order indices, our two nested designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are two nested Latin hypercube designs. First, the block B_ℓ is constructed using the algorithm in [Qia09]. Then, B'_ℓ is obtained by permuting independently the values in each column of B_ℓ . This guarantees that at each step $\ell \geq 0$, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ possess a structure of Latin hypercube and are two replicated designs of order 1. Using the construction proposed in [Qia09], the size of $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ equals at least 2^ℓ .

4.3.2 Nested orthogonal array of strength two

The nested designs constructed for the estimation of closed second-order indices are two nested orthogonal arrays of strength two. Here, a nested orthogonal array of strength two corresponds to an $OA_\lambda(q, d, 2)$ where $\lambda > 1$. It can be partitioned into λ blocks where each block has the geometric structure of an $OA_1(q, d, 2)$. We propose two methods to construct such designs. The first one is an *accept-reject* method. The second one is called *algebraic* method and relies on results from arithmetic.

The idea of each construction is to progressively fill the d -hypercube (the discretized input space) with distinct sub-hypercubes where each sub-hypercube corresponds to a row of a new block. As an illustration, consider the example of an $OA_3(3, 3, 2)$ represented in Figure 4.2 below. This OA is partitioned in three blocks represented in graphs (a), (b), (c). Each block possesses the structure of an $OA_1(3, 3, 2)$. These three blocks form a partition of the hypercube.

FIGURE 4.2 – $OA_3(3, 3, 2)$ with 3 blocks. (a) first block. (b) second block. (c) third block. Each block can be identified as an $OA_1(3, 3, 2)$. Points are represented by their respective sub-hypercube.



The *accept-reject* and the *algebraic* methods strive to construct both an orthogonal array $OA_\lambda(q, d, 2)$ where the rows are two by two distinct. Two rows are said distinct if they differ in at least one component. The idea is to evaluate the model on previously unexplored regions of the input space.

Each method starts with the construction of an initial $OA_1(q, d, 2)$ noted A_0 . Then at each step $\ell \geq 1$, a new $OA_1(q, d, 2)$, noted A_ℓ , is constructed. The two blocks B_ℓ and B'_ℓ are obtained from A_ℓ as in Definition 4.4. As a result, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ both have a structure of $OA_\ell(q, d, 2)$ and are replicated designs of order 2. The process is repeated until the stopping criterion is met. The form of the stopping criterion is discussed in Section 4.4.2.

The *accept-reject* and the *algebraic* methods differ on the way A_ℓ is constructed. We detail below this step for each method.

Method 1 : *accept-reject* Variant 4.1 details the construction of A_ℓ . It uses the operator $\diamond$ defined in Definition 4.7. The idea is to randomly construct a new orthogonal array from A_0 using $\diamond$ and test if its rows are distinct from those of each previous orthogonal array constructed; namely $A_{\ell-1}, A_{\ell-2}, \dots, A_0$. This test may become computationally expensive for small input space dimension as the probability of acceptance decreases faster.

Variant 4.1 *Accept-reject* method for the construction of A_ℓ

```

1: Set bool  $\leftarrow$  false
2: while !bool do
3:   Sample  $\pi_1, \dots, \pi_d$  in  $\Pi_q$ 
4:   Construct  $A_\ell = \diamond(A_0, \{\pi_1, \dots, \pi_d\})$  with (4.7)
5:   for  $k = 0, \dots, \ell - 1$  do
6:     bool $_k \leftarrow \text{rows}(A_\ell) \cap \text{rows}(A_k) == \emptyset$ 
7:   bool  $\leftarrow \forall k : \text{bool}_k$ 

```

Method 2 : *Algebraic method* Define the following set :

$$C = \left\{ g = (0, 0, g_3, \dots, g_d) \mid \forall i \geq 3, g_i \in GF(q) \right\} \subsetneq GF(q)^d.$$

Variant 4.2 details the construction of A_ℓ . $\oplus$ is the addition in $GF(q)^d$. The idea of the method is to construct a partition of the discretized input space and select A_ℓ from this partition. A_ℓ is viewed as a coset of A_0 and is obtained using the set C . The main advantage of this method is that the maximum value taken by ℓ is known beforehand (consequence of Proposition 4.1 thereafter).

Variant 4.2 *Algebraic* method for the construction of A_ℓ

```

1: Choose  $g_\ell \in C$ 
2: Construct  $A_\ell = g_\ell A_0 = \{g_\ell \oplus A_{0i}\}_{i=1}^{q^2}$ ,  $A_{0i} = (A_{0i,1}, \dots, A_{0i,d})$ 
3:  $C \leftarrow C \setminus \{g_\ell\}$ 

```

The following proposition guarantees that A_ℓ constructed in Variant 4.2 is an $OA_1(q, d, 2)$:

Proposition 4.1. *Consider A_0 an $OA_1(q, d, 2)$ based on $GF(q)^d$. We have the following results :*

- i) $\forall g \in GF(q)^d$, gA_0 is an $OA_1(q, d, 2)$
- ii) $\forall g, g' \in C$, such that $g \neq g'$, $gA_0 \cap g'A_0 = \emptyset$. In other words, the sets $\{gA_0\}$ form a partition of $GF(q)^d$.

Proof.

- i) Let $g = (g_1, \dots, g_d) \in GF(q)^d$. Consider A_{0k}, A_{0l} two columns of A_0 . Denote by E the group $(GF(q), +)$. Since $g_k E \times g_l E$ is isomorph to $E \times E$, the 2-tuples $(A_{0i,k} + g_k, A_{0i,l} + g_l)$ obtained after addition are all two by two distinct.
- ii) The proof can be found in [SM95] where an orthogonal array is regarded as a “systematic linear code”.

As a consequence of ii), the maximum number of blocks one can construct using the *algebraic* method equals the cardinality of C , that is q^{d-2} . If this maximum value is reached, the blocks $A_0, A_1, \dots, A_{q^{d-2}-1}$ form a partition of the discretized input space. The cost of our recursive procedure to estimate all closed second-order indices equals $2 \times K \times q^2$ where K is the ending step.

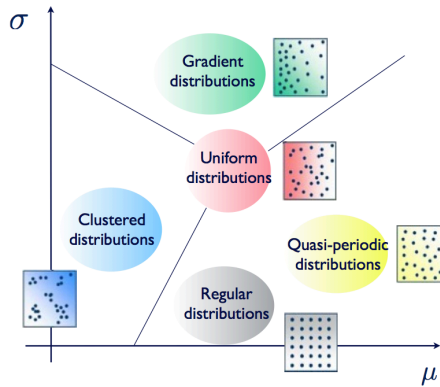
4.4 Space-filling properties and application

We propose first to study the space-filling properties of the nested design used in our recursive procedure. Then, an application of our recursive procedure is presented on a toy example.

4.4.1 Space-filling properties

Three criteria are selected to study the properties of the nested designs : the maximin [JMY90], the emst (euclidean minimal spanning tree [FVCS09]) and the L^2 star discrepancy [Hic98a]. The maximin criterion returns the minimum of the distances between all pairs of points of a design. It can be interpreted as follows : the higher the value, the more regular the scattering of design points. The emst criterion can be interpreted using a (μ, σ) graph, Figure 4.3, called interpretation graph.

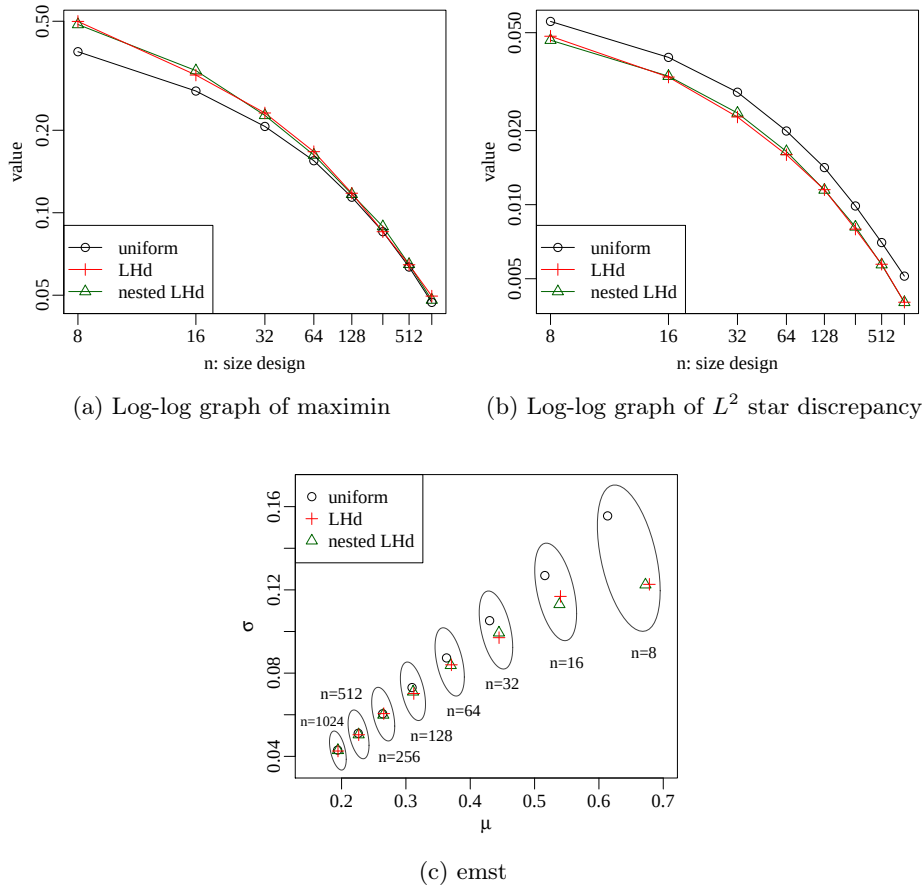
FIGURE 4.3 – Interpretation graph of the emst. The uniform distribution is used as a reference.



A minimal spanning tree is constructed from the design, then mean (μ) and standard deviation (σ) of the tree edges lengths are evaluated. A value of the emst criterion is represented as a point in the (μ, σ) graph. The uniform distribution, that is i.i.d sampling, is used as a reference. A design having a higher value for μ and a smaller value for σ than those of a uniform design is more regular. Maximin and emst criteria provide together a good estimation of the regularity properties of a design. The L^2 star discrepancy criterion measures the uniformity property of a design. The smaller the value, the more uniform is the design.

We first study properties of the nested Latin hypercube design (nested LHd) used for the estimation of first-order Sobol' indices. Its properties are compared to those of the following designs : (i) uniform design (obtained through i.i.d sampling) and (ii) Latin hypercube design (LHd). Figure 4.4 shows the results obtained with each of the three criteria. The results are averaged over $r = 100$ repetitions.

FIGURE 4.4 – Averaged results of maximin, emst and star discrepancy criteria over 100 repetitions for different sizes n of the designs used for the estimation of first-order indices.



The input space dimension d equals 5. The comparison is made for the following sizes n of each design : $(2^3, 2^4, \dots, 2^{10})$. For the nested LHd, these sizes correspond to those of design $\mathcal{P}_\ell$ augmented over 8 consecutive steps. Both the LHd and the nested LHd give similar results for the three criteria. Furthermore, both designs give better results than the uniform design. As such, in terms of space-filling properties of the designs, there is no drawback to render the replication procedure recursive.

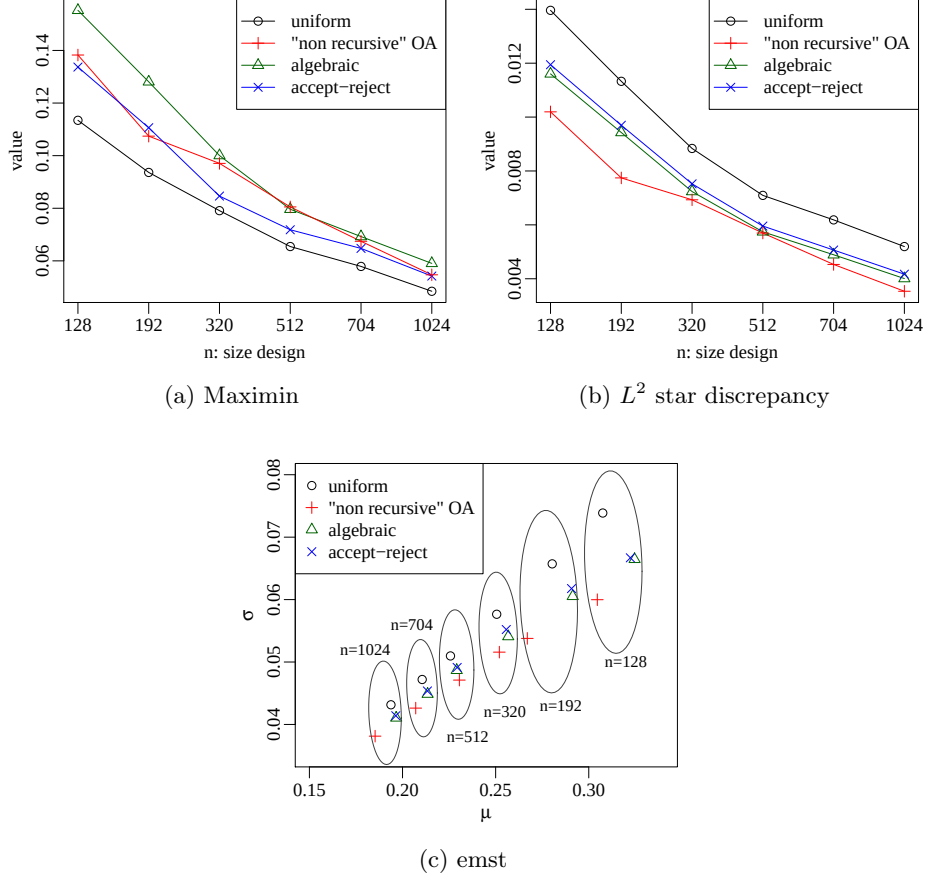
Remark 4.2. *One other class of designs well suited for the estimation of first-order Sobol' indices are low discrepancy sequences. These sequences are points sets sampled so as to approximate as close as possible a uniform distribution and are known to achieve both uniformity and regularity properties. Such sequences could be used in our recursive procedure in place of nested Latin hypercube designs. This alternative is detailed in Chapter 5.*

A second comparison is carried out between the following designs used for the estimation of closed-second order indices : (i) uniform design, (ii) “non-recursive” OA, (iii) *accept-reject* and (iv) *algebraic*. Design (ii) refer to the orthogonal array used in [TP15]. Designs (iii) and (iv) refers to the design $\mathcal{P}_\ell$ constructed with either the *accept-reject* or the *algebraic* method. Results are again averaged over $r = 100$ repetitions and the input space dimension still equals 5. Figure 4.5 shows the results obtained with each of the three criteria.

For the sake of visualization, results for only the following sizes n of the designs are represented : $(3 \times 8^2, 5 \times 8^2, 8 \times 8^2, 11 \times 8^2, 15 \times 8^2, 18 \times 8^2)$. In terms of emst and discrepancy criteria, the “non-recursive” OA gives the best results while results for the *accept-reject* and *algebraic* designs are similar. The *algebraic* design gives better results for the maximin criterion than the *accept-reject* design.

The main conclusion is that the *algebraic* design possesses regularity and uniformity properties overall slightly better than those of the *accept-reject* design. These two designs possess slightly worse space-filling properties than their counterpart used in [TP15]. This difference can be explained by the lack of progressive discretization of the inputs in both the *algebraic* and the *accept-reject* method. However, that is largely offset by the possibility to perform a recursive estimation of the indices.

FIGURE 4.5 – Averaged results of maximin, emst and star discrepancy criteria over 100 repetitions for different sizes n of the designs used for the estimation of closed second-order indices.



4.4.2 Application to a toy example

Our recursive procedure is tested and compared to the classic replication procedure with the Bratley *et al.* function [BFN92], defined as follows :

$$f(X_1, \dots, X_d) = \sum_{i=1}^d (-1)^i \prod_{k=1}^i X_k ,$$

where $X_1, \dots, X_d$ are independent random variables uniformly distributed on $[0, 1]$. Both first- and closed second-order Sobol' indices of the function are estimated with each procedure. Both procedure are repeated $r = 100$ times to get samples of estimates. We choose $d = 6$ for the input space dimension. Since f has an analytical expression, theoretical values of the Sobol' indices can be precisely calculated through symbolic integrals evaluations.

Stopping criterion Our recursive procedure is carried out until a stopping criterion is reached. At each step ℓ of the procedure, the following quantity is evaluated :

$$\mathbf{e}^{(\ell)} = \left\| \hat{\underline{S}}_u^{(\ell)} - \hat{\underline{S}}_u^{(\ell-1)} \right\|,$$

where $\|\cdot\|$ denotes the absolute value function. $\mathbf{e}^{(\ell)}$ is an absolute difference between two successive estimations of $\underline{S}_u$. The stopping criterion we proposed is composed of two conditions c_1 and c_2 . The first condition c_1 reads as follows :

$$\forall u \in \mathcal{D} : \mathbf{e}^{(\ell-\ell_0)} < \varepsilon, \mathbf{e}^{(\ell-\ell_0-1)} < \varepsilon, \dots, \mathbf{e}^{(\ell)} < \varepsilon$$

where $\mathcal{D}$ equals either $\{1, \dots, d\}$ or $\{(k, l) \in \{1, \dots, d\}^2; k < l\}$ depending on whether first-order or closed second-order Sobol' indices are estimated and $\ell_0 > 0$ is an integer. Condition c_1 tests if all quantities $\mathbf{e}^{(\ell)}$ are smaller than a tolerance ε on ℓ_0 consecutive steps. The second condition c_2 tests if $\ell > \ell_{max}$, where ℓ_{max} is a maximum number of iterations. The parameters ε , ℓ_0 and ℓ_{max} have to be properly set.

The recursive procedure stops when one of the two conditions is verified. From the r repetitions, a vector $(r_1, \dots, r_K, \dots, r_{\ell_{max}})$ is constructed where r_K denotes the number of time our recursive procedure has stopped at step K and $r_{\ell_{max}}$ denotes the number of times condition c_2 has been reached. We note by r_α the median of this vector and α the corresponding step.

To have a fair comparison, $\underline{S}_u$ is also estimated r times with the classic replication procedure where the size of the two replicated designs equals the size of $\mathcal{P}_\alpha$ and $\mathcal{P}'_\alpha$ in our recursive procedure.

Estimation of first-order indices We consider the context where a limited number of evaluation points is available as it is often the case in industrial applications. Therefore, a small value for ℓ_{max} is selected to highlight that our recursive procedure can perform as well as the classic one for a restricted budget of evaluation points. The parameters of the stopping criterion for the recursive procedure are set as follows : $\varepsilon = 0.15$, $\ell_0 = 2$ and $\ell_{max} = 9$. The size of designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ is instantiated at 2^2 and can raise up to $2^{\ell_{max}}$.

Figure 4.6 shows a barplot representation of the r_K obtained. We observe that condition c_2 is only reached one third of the time. Figure 4.7 shows the boxplots representation of the estimates obtained with the two replication procedures : recursive (right boxplots) and classic (left boxplots).

The two methods give overall similar results. Hence, there is no drawback to render the replication procedure recursive for the estimation of first-order indices. Furthermore, our recursive procedure shows that the number of model evaluations can be decreased by adopting a sequential approach. One can calculate the gain in terms of number of evaluations. This gain corresponds to the ratio of the maximum number of evaluations $r_{\ell_{max}}$ divided by the iteration at which our recursive procedure

stopped. For this example the median gain equals $9/8 = 1.125$ and the maximum gain equals $9/7 = 1.29$.

FIGURE 4.6 – Distribution of the r_K for the estimation of first-order indices. The bar associated to the step α is colored in black.

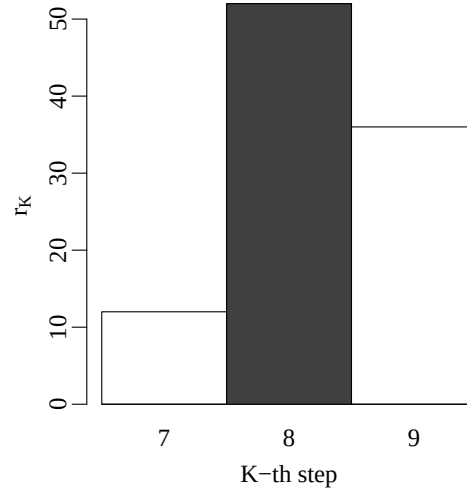
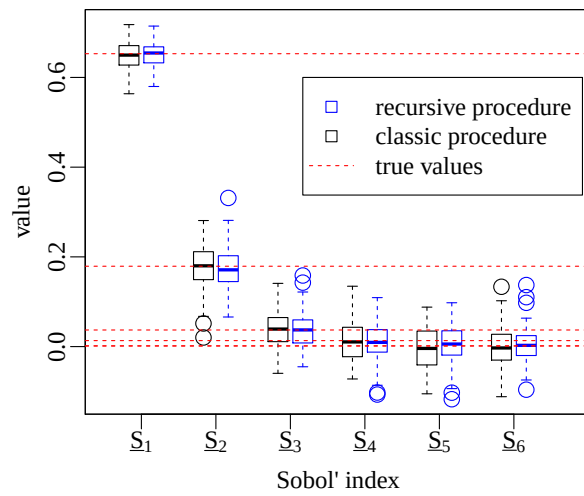


FIGURE 4.7 – Boxplots of first-order Sobol' indices estimated $r = 100$ times with both our recursive procedure (left boxplot) and the classic replication procedure (right boxplot). The dotted horizontal lines refer to the true values of the indices. True values of indices $\underline{S}_5$ and $\underline{S}_6$ are identical.



Estimation of closed second-order indices The parameters of the stopping criterion are set as follows : $\varepsilon = 3 \times 10^{-3}$, $\ell_0 = 3$ and $\ell_{max} = 100$. The initial orthogonal array A_0 used to augment designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ is constructed by setting $q = 8$. The sizes of these designs can range from 8^2 up to 100×8^2 .

Figure 4.8 shows barplots representation of the r_K obtained when applying our recursive procedure with either the *algebraic* method or the *accept-reject* method. Results show that our recursive procedure finishes at earlier steps when using the *algebraic* method.

FIGURE 4.8 – Distribution of the r_K when our recursive procedure is applied with either (a) the *algebraic* method or (b) the *accept-reject* method. For each graph, the bar associated to the median step α is colored in black.

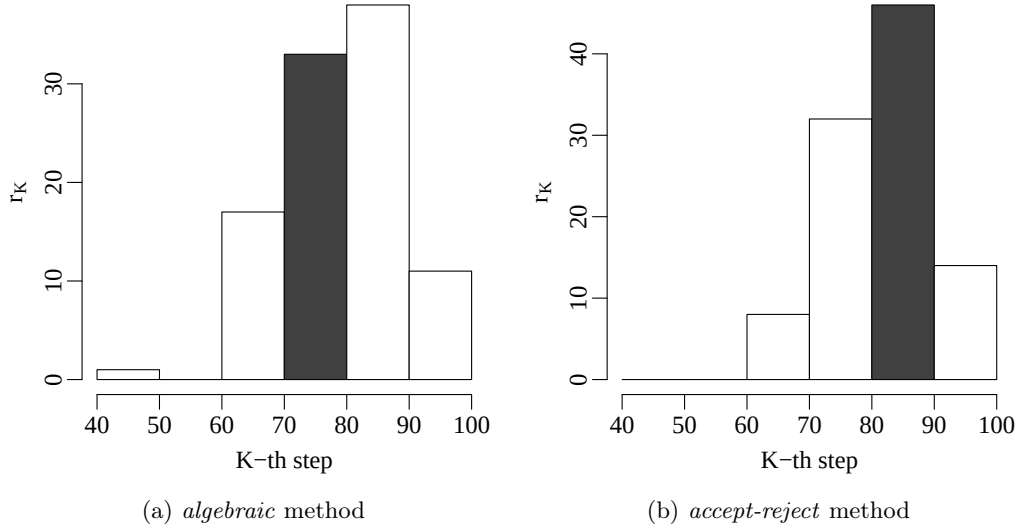
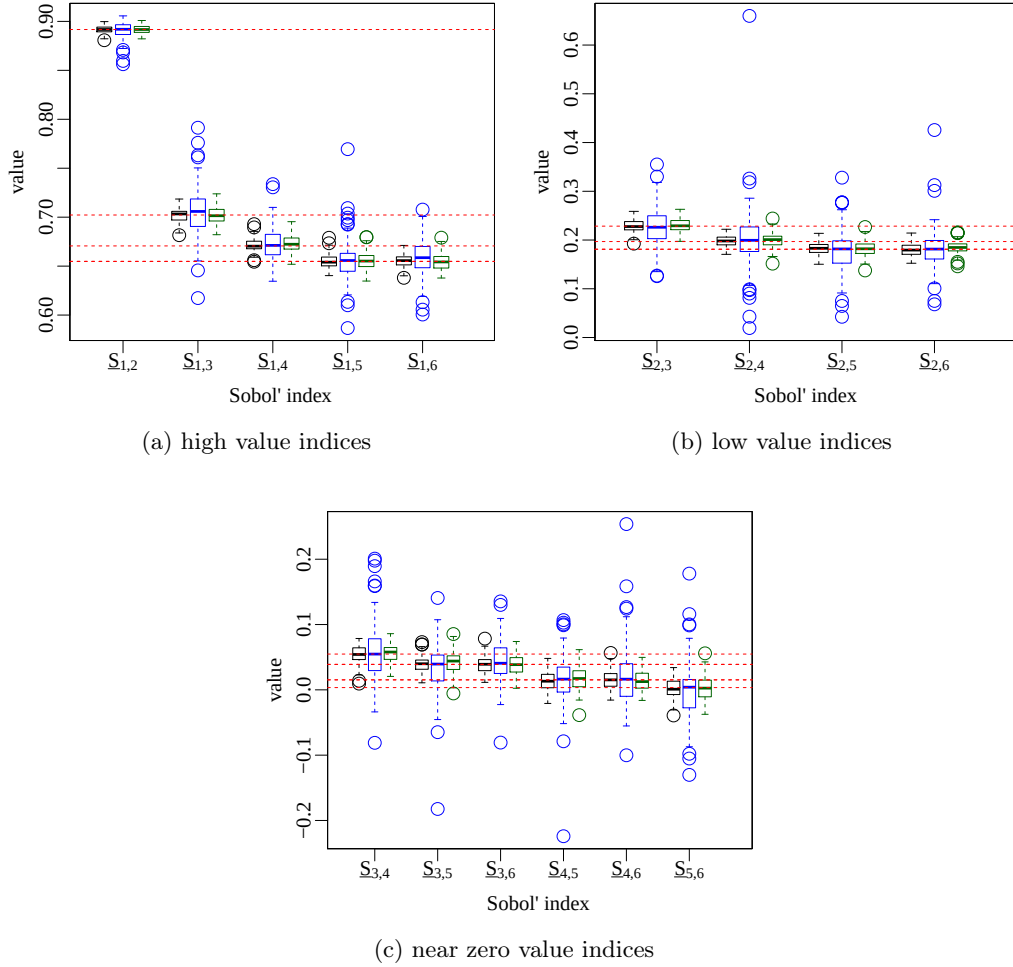


Figure 4.9 gives the boxplots representation of the estimates obtained with the classic replication procedure (left boxplots) and with our recursive procedure using either the *algebraic* method (middle boxplots) or the *accept-reject* method (right boxplots).

The main observation is that our recursive procedure using the *algebraic* method shows more variability in the estimates than the two others. This observation is emphasized for graphs (b) and (c) of Figure 4.9 corresponding to Sobol' indices with low values. However, this variability is mostly due to the *algebraic* method itself stopping at earlier steps than the *accept-reject* method. The results obtained with our recursive procedure using the *accept-reject* method are overall similar to those obtained with the classic replication procedure.

As for the case of first-order indices, one can calculate the gain of our recursive procedure in terms of number of evaluations. Table 4.1 gives the gain of our method for each quartile of the vector $(r_1, \dots, r_{\ell_{max}})$. Our recursive procedure shows that it

FIGURE 4.9 – Boxplots of closed second-order Sobol' indices estimated $r = 100$ times with our recursive procedure and the classic replication procedure. For each index $\underline{S}_u$, the left boxplot refers to the classic replication procedure, the boxplot in the middle (resp. on the right) refers to our recursive procedure using the *algebraic* (resp. *accept-reject*) method. The horizontal dotted lines refer to the true values of the indices.



is possible to decrease even more the number of simulations by adopting a sequential approach for the estimation of closed second-order indices while preserving the same order of precision. However, as stated before, there is a computational price to pay induced by the *accept-reject* method. When the input space dimension is small ($d \leq 4$), it is harder to find new blocks. As such, the *algebraic* method should be preferred to the *accept-reject* one. At the opposite, when the input space dimension is high, new blocks are easier to find. Therefore, the *accept-reject* method should be used as it gives more accurate results.

TABLE 4.1 – Gain of the recursive replication procedure using either the *algebraic* or the *accept-reject* construction. The gain is calculated in terms of number of evaluations for each quartile $(r_{1/4}, r_{1/2}, r_{3/4})$ of the vector $(r_1, \dots, r_{\ell_{max}})$.

quartile	construction	value	gain = $\frac{r_{\ell_{max}}}{\text{value}}$
$r_{1/4}$	<i>algebraic</i>	73	1.37
	<i>accept-reject</i>	76	1.32
$r_{1/2}$	<i>algebraic</i>	80	1.25
	<i>accept-reject</i>	82	1.22
$r_{3/4}$	<i>algebraic</i>	87	1.15
	<i>accept-reject</i>	88	1.14

Conclusion

In this paper we proposed a new approach rendering the replication procedure recursive to estimate first-order or closed second-order Sobol' indices. We introduced a recursive formula for the Sobol' index estimator. The recursive procedure presented consists in augmenting the two replicated designs with new sets of points through the construction of nested space-filling designs. For the case of closed second-order indices, two methods were proposed to construct a nested orthogonal array of strength two : an *algebraic* method and an *accept-reject* method. Our recursive procedure was compared to the classic replication procedure of Tissot and Prieur [TP15]. The comparison focused on the space-filling properties of the designs and on the precision of the Sobol' indices estimates.

The replication procedure proposed in [TP15] are known to be highly efficient in terms of number of simulations. Yet the results in this paper showed that it is still possible to decrease the number of simulations by adopting a sequential procedure based on a recursive method of estimation. More precisely, the nested designs proposed here gave the same order of precision on Sobol' indices as the replicated designs used in [TP15] but with a random number of simulations of much smaller expectation. Furthermore, the space-filling properties of the nested designs constructed were on average as good as the one of the replicated designs used in [TP15].

For the case of first-order indices, considering Sobol' sequences could improve the nested designs (see constructions of Chapter 5). For the case of closed second-order indices, the variability in the results showed by our recursive procedure while using the *algebraic* method could be reduced by further working on the set C (Section 4.3.2 Variant 2). In our case, the set C was filled with elements g chosen at random. A more deterministic choice of the g could lead to a better exploration of the input space.

Acknowledgments

This work is supported by the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020).

CHAPITRE 5

Reliable error estimation for Sobol' indices

Les résultats de ce chapitre ont donné lieu à un compte-rendu mathématique de l'Académie des Sciences [GRA⁺16] et un article[GR16], soumis pour publication. L'objectif de ce chapitre est d'apporter une réponse à la problématique P_3 soulevée en introduction portant sur la construction d'une borne d'erreur afin de garantir un seuil de précision pour l'estimation des indices de Sobol'. Dans ce chapitre les plans d'expériences sont construits en utilisant des séquences de Sobol'.

Les sections 5.1 à 5.4 constituent le compte-rendu mathématique [GRA⁺16]. Dans ce compte-rendu, on se place dans le contexte de l'estimation des effets principaux des paramètres d'un modèle. A cet effet, nous proposons d'étudier deux approches pour construire itérativement des plans répliqués à partir de séquences de Sobol'. Les propriétés d'uniformité et de remplissage de l'espace des plans construits sont étudiées sur la base de deux critères.

Les sections 5.5 à 5.8 constituent l'article [GR16]. Dans cet article, nous proposons une borne d'erreur pour l'estimation d'indices de Sobol' à l'aide de séquences de Sobol'. La borne d'erreur est obtenue à partir d'une décomposition en coefficients de Walsh des intégrandes intervenant dans la formule intégrale de l'indice de Sobol'. Cette décomposition utilise la propriété de clôture des séquences de Sobol'. Nous proposons d'utiliser cette borne d'erreur comme critère d'arrêt pour deux méthodes d'estimation des indices : méthode de Saltelli et méthode répliquée.

Notons que cette borne d'erreur ne peut pas être utilisée pour la version récursive de la méthode répliquée présentée au chapitre 4. En effet, les plans emboîtés construits pour cette dernière méthode ne satisfont pas la propriété de clôture vérifiée par les séquences de Sobol'.

Titre : Iterative construction of replicated designs based on Sobol' sequences.

Auteurs : Laurent Gilquin^{a,b}, Lluís Antoni Jiménez Rugama^c, Elise Arnaud^b, Fred J. Hickernell^c, Hervé Monod^d, Clémentine Prieur^b.

a Inria Grenoble - Rhône-Alpes, Inovalée, 655 avenue de l'Europe, 38330 Montbonnot

b Univ. Grenoble Alpes, Jean Kuntzmann Laboratory, F-38000 Grenoble, France

CNRS, LJK, F-38000 Grenoble, France, Inria

c Illinois Institute of Technology, Rettaliata Engineering Center, 10 W 32st, Chicago, IL 60616

d MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-En-Josas, France

5.1 Introduction

Mathematical models often involve a substantial number of poorly known parameters. The effect of these parameters on the output of the model can be assessed through sensitivity analysis. Global sensitivity analysis methods are useful tools to identify the parameters having the most influence on the output. A well known approach is the variance based method introduced by Sobol' in [Sob93]. This method estimates sensitivity indices called Sobol' indices that summarize the influence of each model input. Among all Sobol' indices, one can distinguish the first-order indices that estimate the main effect of each input.

The procedure to estimate first-order Sobol' indices proposed by Sobol' and its improvements (see Saltelli [SCS08] for an exhaustive survey) all suffer from a prohibitive number of model evaluations that grows with respect to the input space dimension. An elegant solution to reduce this number relies on the construction of particular designs of experiments called replicated designs. The notion of replicated designs was first introduced by McKay through its definition of replicated Latin Hypercubes in [McK95]. Below we provide this definition in a more general framework :

Definition 5.1. Consider $\mathbf{x} \in [0, 1]^s$, and $\mathbf{x}_u \in [0, 1]^{|u|}$ the subset of elements of $\mathbf{x}$ given by $u \subsetneq \{1, \dots, s\}$, where $|u|$ is the cardinality of u . Let $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$ and $\mathcal{P}' = \{\mathbf{x}'_i\}_{i=0}^{n-1}$ be two point sets in $[0, 1]^s$, and denote by $\mathcal{P}^u = \{\mathbf{x}_{i,u}\}_{i=0}^{n-1}$ (resp. $\mathcal{P}'^u$) the subset of elements of points in $\mathcal{P}$ (resp. $\mathcal{P}'$) indexed by u . We say that $\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order $a \in \{1, \dots, s-1\}$, if for any $u \subsetneq \{1, \dots, s\}$ with $|u| = a$, $\mathcal{P}^u$ and $\mathcal{P}'^u$ are the same point set in $[0, 1]^a$, perhaps in a different order.

The replication procedure described in [MJ08, TP15] allows the estimation of all first-order Sobol' indices with only two replicated designs of order 1. This procedure has the major advantage of reducing the number of model evaluations, evaluating only on designs $\mathcal{P}$ and $\mathcal{P}'$ regardless of the input space dimension. However, Sobol'

indices estimates may still not be accurate enough if designs $\mathcal{P}$ and $\mathcal{P}'$ do not explore the input space properly.

In this note, we propose two different constructions of replicated designs of order 1 based on Sobol' sequences. Both constructions ensure that the input space is properly explored and can be used within the replication procedure to estimate all first-order Sobol' indices. The definition of these constructions is recursive, therefore one can iteratively refine each replicated design by adding the corresponding new set of points. We first provide a brief introduction on digital sequences and then present two iterative approaches to construct the two replicated point sets. We end this note by analyzing the space-filling properties of the two designs constructed.

5.2 Digital sequences background

5.2.1 Preliminaries

Digital nets and sequences were first introduced by Niederreiter [Nie92b] in the numerical integration framework to define good uniformly distributed points in $[0, 1]^s$. They can also appear in the literature as digital (t, m, s) -nets and digital (t, s) -sequences, or simply (t, m, s) -nets and (t, s) -sequences. Sobol' and Niederreiter-Xing sequences are two examples of digital sequences detailed in [Sob93] and [NX96].

Definition 5.2. Let $\mathcal{A}$ be the set of all elementary intervals $A \subset [0, 1]^s$ where $A = \prod_{j=1}^s [\alpha_j b^{-\gamma_j}, (\alpha_j + 1)b^{-\gamma_j})$, with integers $s \geq 1$, $b \geq 2$, $\gamma_j \geq 0$, and $b^{\gamma_j} > \alpha_j \geq 0$. For $m \geq t \geq 0$, the point set $\mathcal{P} \in [0, 1]^s$ with b^m points is a (t, m, s) -net in base b if every A with volume b^{t-m} contains b^t points of $\mathcal{P}$.

Thus, a (t, m, s) -net is defined such that all elementary intervals of volume b^{t-m} will enclose the same proportion of points of $\mathcal{P}$, namely $b^{t-m}|\mathcal{P}|$ points. The most evenly spread nets are $(0, m, s)$ -nets, since each elementary interval of the smallest volume possible, b^{-m} , contains exactly one point. The quality of any (t, m, s) -net is therefore measured by the parameter t , called t -value.

By increasing m , we increase the number of points of the (t, m, s) -net. In the limiting case where $m \rightarrow \infty$, we can define the (t, s) -sequence as :

Definition 5.3. For integers $s \geq 1$, $b \geq 2$, and $t \geq 0$, the sequence $\{\mathbf{x}_i\}_{i \in \mathbb{N}_0}$ is a (t, s) -sequence in base b , if for every set $\mathcal{P}_{\ell, m} = \{\mathbf{x}_i\}_{i=\ell b^m}^{(\ell+1)b^m-1}$ with $\ell \geq 0$ and $m \geq t$, $\mathcal{P}_{\ell, m}$ is a (t, m, s) -net in base b .

The replicated design properties can also apply to digital sequences. Hence, we introduce the following definition,

Definition 5.4. Two digital sequences $\{\mathbf{x}_i\}_{i \in \mathbb{N}_0}$ and $\{\mathbf{x}'_i\}_{i \in \mathbb{N}_0}$ are digitally replicated of order a if for all $m \geq 0$, $\{\mathbf{x}_i\}_{i=0}^{b^m-1}$ and $\{\mathbf{x}'_i\}_{i=0}^{b^m-1}$ are two replicated designs of order a .

5.2.2 Sobol' sequences

Sobol' sequences in dimension s are digital sequences in base $b = 2$ that can be computed using the *generating matrices*, a set of s full rank infinite dimensional upper triangular matrices over the Galois field $\mathbb{F}_2 := \{0, 1\}$. These generating matrices are recursively constructed given some primitive polynomials and initial directional numbers. In [JK08], Kuo and Joe detail this construction and also suggest a particular choice for these matrices that optimize the 2 dimensional projection t -values.

Consider the generating matrices $C_1, \dots, C_s$, and $C_1^m, \dots, C_s^m$ their upper left corner blocks of size $m \times m$. Although $C_1, \dots, C_s$ are of infinite size, one only requires the knowledge of $C_1^m, \dots, C_s^m$ to construct the first 2^m Sobol' points : for each $i = 0, \dots, 2^m - 1$, the point $\mathbf{x}_i = (x_{i,1}, \dots, x_{i,s})^\top$ of the sequence is obtained dimension-wise by :

$$(x_{i,j,1}, \dots, x_{i,j,m})^\top = C_j^m \mathbf{i}, \quad j = 1, \dots, s, \quad (5.1)$$

where $x_{i,j} = \sum_{k \geq 1} x_{i,j,k} 2^{-k}$ is the binary expansion of $x_{i,j}$ and $\mathbf{i} = (i_0, \dots, i_{m-1})^\top$ is the vector obtained from the binary expansion of $i = \sum_{k \geq 0} i_k 2^k$. All matrix operations are performed in $\mathbb{F}_2$. Below we provide an example of how to compute $x_{11,1} = 0.8125$ and $x_{9,2} = 0.4375$,

$$\begin{pmatrix} x_{11,1,1} \\ x_{11,1,2} \\ x_{11,1,3} \\ x_{11,1,4} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{C_1^4} \underbrace{\begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}}_{11} = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \begin{pmatrix} x_{9,2,1} \\ x_{9,2,2} \\ x_{9,2,3} \\ x_{9,2,4} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{C_2^4} \underbrace{\begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}}_9 = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

To compute the next 2^m points of the sequence, one can infer from (5.1) that

$$(x_{i+2^m,j,1}, \dots, x_{i+2^m,j,m+1})^\top = (x_{i,j,1}, \dots, x_{i,j,m}, 0)^\top \oplus (c_j^{m+1})^\top, \quad i = 0, \dots, 2^m - 1, \quad (5.2)$$

where c_j^{m+1} is the last column of C_j^{m+1} and $\oplus$ is the addition in $\mathbb{F}_2$.

In addition, Sobol' sequences have good group structure properties. For any $m \geq 1$, the first 2^m points of the sequence form an Abelian group under $\oplus$:

$$\begin{aligned} \mathbf{x} &= \left(\sum_{k=1}^m x_{1,k} 2^{-k}, \dots, \sum_{k=1}^m x_{s,k} 2^{-k} \right)^\top, & \mathbf{z} &= \left(\sum_{k=1}^m z_{1,k} 2^{-k}, \dots, \sum_{k=1}^m z_{s,k} 2^{-k} \right)^\top, \\ \mathbf{x} \oplus \mathbf{z} &:= \left(\sum_{k=1}^m (x_{1,k} + z_{1,k} \bmod 2) 2^{-k}, \dots, \sum_{k=1}^m (x_{s,k} + z_{s,k} \bmod 2) 2^{-k} \right)^\top. \end{aligned} \quad (5.3)$$

From equation (5.1) above, one may see that the first 2^m points of the sequence are elements of $(\mathbb{F}_2^m)^s$.

Lemma 5.1. *All Sobol' sequences are digitally replicated of order 1.*

Proof : Consider any two s -dimensional Sobol' sequences generated by $C_1, \dots, C_s$ and $C'_1, \dots, C'_s$ respectively. Since any two matrices C_j^m and $C_j'^m$ are square and full rank, the operations $\mathbf{i} \mapsto C_j^m \mathbf{i}$ and $\mathbf{i} \mapsto C_j'^m \mathbf{i}$ are one-to one and onto for all $i = 0, \dots, 2^m - 1$. Therefore, they generate the same point sets.

5.3 Iterative constructions of replicated point sets

In this section we propose two different approaches to iteratively construct two replicated point sets, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$, based on Sobol' sequences. These two constructions are carried out according to the following recursive scheme :

$$\begin{cases} \mathcal{P}_0 = B_0 \\ \mathcal{P}_\ell = \mathcal{P}_{\ell-1} \cup B_\ell \end{cases} \quad \begin{cases} \mathcal{P}'_0 = B'_0 \\ \mathcal{P}'_\ell = \mathcal{P}'_{\ell-1} \cup B'_\ell \end{cases}, \quad \ell \geq 1,$$

where B_ℓ and B'_ℓ are new sets of points added at step ℓ to refine $\mathcal{P}_{\ell-1}$ and $\mathcal{P}'_{\ell-1}$. For all $\ell \geq 0$, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are two replicated designs of order 1.

The first approach is called multiplicative because $|\mathcal{P}_\ell| = 2^\ell$ while the second one is called additive and $|\mathcal{P}_\ell| = \ell|B_0|$. In the multiplicative case, we will directly use 2 s -dimensional sequences as a result from Lemma 5.1. However, for the additive case, we will consider an initial set of Sobol' points and apply different scramblings and digital shifts to extend the point sets. Additionally, in both cases one can randomize the points using Owen's scrambling [Owe95] as long as same coordinates of $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ share the same scrambling.

5.3.1 Multiplicative approach

Two replicated point sets of order 1, $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$, can be constructed using two s -dimensional Sobol' sequences. We note $C_1, \dots, C_s$ the generating matrices used to generate $\mathcal{P}_\ell$ and $C'_1, \dots, C'_s$ those used to generate $\mathcal{P}'_\ell$. To ensure that $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are as uniform as possible, these generating matrices need to be different from each other. We choose $C_1, \dots, C_s, C'_1, \dots, C'_s$ to be the first $2s$ generating matrices suggested by Joe and Kuo in [JK08], not necessarily in this order. In [JK08], Joe and Kuo minimized the t -value for all 2 dimensional projections.

In order to iteratively extend designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$, one just needs to compute $B_\ell = \{\mathbf{x}_{2^{\ell-1}}, \dots, \mathbf{x}_{2^\ell-1}\}$ and $B'_\ell = \{\mathbf{x}'_{2^{\ell-1}}, \dots, \mathbf{x}'_{2^\ell-1}\}$, starting with $B_0 = B'_0 = \{\mathbf{0}\}$, with $\mathbf{0}$ the null vector. Each set B_ℓ and B'_ℓ can be constructed using (5.2) applied to $\mathcal{P}_{\ell-1}$ and $\mathcal{P}'_{\ell-1}$.

As a direct consequence of Lemma 5.1, at each step ℓ designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are two replicated designs of order 1. Furthermore, they both inherit the space-filling properties of (t, ℓ, s) -nets.

5.3.2 Additive Approach

With the multiplicative approach, the size of designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ is multiplied by 2 each time ℓ is increased by one. This growth rate may be inadequate for some applications. The additive approach presented in this section is attractive due to a slower size growth. Given an initial choice of $r \geq 1$ that specifies the size of B_0 and B'_0 , only $|B_0| = 2^r$ points are added to both designs at each step. The main drawback of this approach is that $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ do not inherit the structure of a Sobol' sequence when $\ell \geq 1$. Nevertheless, both designs display good space-filling properties as shown in the next section.

Analogously to the multiplicative case, the two replicated point sets $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ constructed with the additive approach are iteratively refined with B_ℓ and B'_ℓ .

First $\{\mathbf{x}_i\}_{i=0}^{2^r-1}$ and $\{\mathbf{x}'_i\}_{i=0}^{2^r-1}$ are set to be the first 2^r points of two s -dimensional Sobol' sequences. The generating matrices of these two sequences are selected as in the multiplicative approach. Then, B_ℓ (resp. B'_ℓ), for $\ell \geq 0$, is obtained from $\{\mathbf{x}_i\}_{i=0}^{2^r-1}$ (resp. $\{\mathbf{x}'_i\}_{i=0}^{2^r-1}$) by carrying out digital shifts and scrambling operations. Therefore, both B_ℓ and B'_ℓ inherit the (t, r, s) -net structure of these initial sets.

At step $\ell = 0$, $B_0 = \{\mathbf{x}_i^{(0)}\}_{i=0}^{2^r-1}$ and $B'_0 = \{\mathbf{x}'_i^{(0)}\}_{i=0}^{2^r-1}$ are generated as follows : for each $i = 0, \dots, 2^r - 1$, points $\mathbf{x}_i^{(0)} = (x_{i,1}^{(0)}, \dots, x_{i,s}^{(0)})^\top$ and $\mathbf{x}'_i^{(0)} = (x'_{i,1}^{(0)}, \dots, x'_{i,s}^{(0)})^\top$ are obtained by linearly transforming $\mathbf{x}_i$,

$$\begin{aligned} (x_{i,j,1}^{(0)}, \dots, x_{i,j,r}^{(0)})^\top &= L(x_{i,j,1}, \dots, x_{i,j,r})^\top, & j = 1, \dots, s, \\ (x'_{i,j,1}^{(0)}, \dots, x'_{i,j,r}^{(0)})^\top &= L'(x_{i,j,1}, \dots, x_{i,j,r})^\top, & j = 1, \dots, s, \end{aligned} \quad (5.4)$$

where $x_{i,j}^{(0)} = \sum_{k=1}^r x_{i,j,k}^{(0)} 2^{-k}$, $x'_{i,j}^{(0)} = \sum_{k=1}^r x'_{i,j,k}^{(0)} 2^{-k}$ and L, L' are two distinct full rank lower triangular matrices of size $r \times r$ over $\mathbb{F}_2$. Left multiplications by L and L' in (5.4) are called linear scramblings [Mat98].

At step $\ell \geq 1$, using addition as defined in (5.3) we construct :

$$B_\ell = \{\mathbf{x}_i^{(0)} \oplus \mathbf{e}_\ell\}_{i=0}^{2^r-1}, \quad B'_\ell = \{\mathbf{x}'_i^{(0)} \oplus \mathbf{e}'_\ell\}_{i=0}^{2^r-1}. \quad (5.5)$$

where $\mathbf{e}_\ell \in (\mathbb{F}_2^r)^s \setminus \mathcal{P}_{\ell-1}$ and $\mathbf{e}'_\ell \in (\mathbb{F}_2^r)^s \setminus \mathcal{P}'_{\ell-1}$. Indeed, B_0 is a subgroup of $(\mathbb{F}_2^r)^s$. Therefore, we construct $B_\ell = B_0 \oplus \mathbf{e}_\ell$ as a coset of B_0 under the right choice of $\mathbf{e}_\ell$. The same reasoning applies to B'_ℓ . Additions by vectors $\mathbf{e}_j$ and $\mathbf{e}'_j$ in (5.5) are referred as digital shifts.

There are exactly 2^{rs} different choices of $\mathbf{e}_\ell$ and $\mathbf{e}'_\ell$ in $(\mathbb{F}_2^r)^s$, and $2^{r(r-1)/2}$ choices of L and L' . The maximum number of iterations ℓ that can be performed equals the number of cosets, $2^{r(s-1)}$. If this maximum is reached, designs $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ both correspond to a full grid design of 2^{rs} points with 2^r distinct values per input.

Lemma 5.2. *Under the additive construction $\mathcal{P}_\ell$ and $\mathcal{P}'_\ell$ are two replicated designs of order 1, $\ell \geq 0$.*

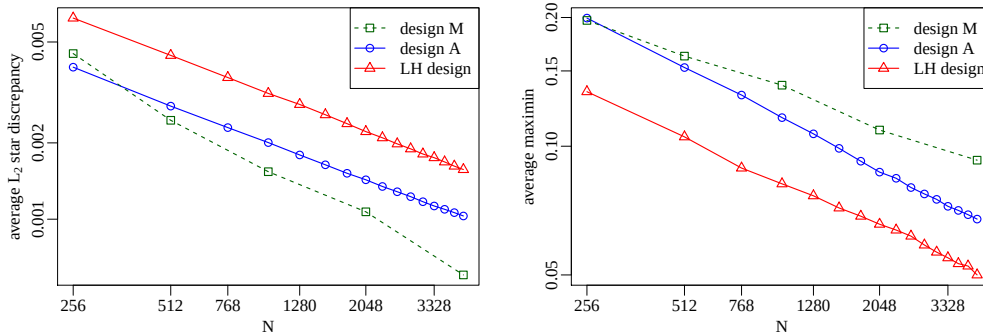
Proof : For each $\ell \geq 1$, digital shift and scrambling operations carried out in equations (5.4) and (5.5) are bijections from $\mathbb{F}_2^r$ and $(\mathbb{F}_2^r)^s$. Hence, the same coordinates of B_ℓ and B'_ℓ will contain the same set of 2^r values.

5.4 Space-filling properties

To analyze the space-filling properties of both approaches we will use two criteria. The first one is the L_2 star discrepancy that measures the uniformity of a point set. This criterion can be easily computed through an analytical expression provided by [MC94]. The lower the criterion, the more uniform the point set is. Sobol' sequences were originally constructed to be low discrepancy sequences and therefore, behave well under this criterion. The second criterion is called maximin distance and measures the regularity of a point set. It returns the minimum of all Euclidean distances between pairs of points in the set. The higher the criterion, the more spread the point set is.

We name by design M (resp. A) one of the two replicated designs constructed with the multiplicative (resp. additive) approach, and set the input space dimension s to 6. For design M , we study the properties of $\mathcal{P}_8, \dots, \mathcal{P}_{12}$, and for design A , we set the initial value to $r = 8$ and study the properties of $\mathcal{P}_0, \dots, \mathcal{P}_{15}$. For both M and A we generate 100 designs whose average criteria are presented in Figure 5.1. While each of the 100 designs M is independently randomized applying Owen's scrambling [Owe95], designs A are constructed by randomly selecting matrix L (uniformly among all $2^{r(r-1)/2}$ choices) and vectors $\mathbf{e}_\ell$ used in equation (5.5). In addition, we also show in Figure 5.1 the results obtained with a Latin Hypercube design (LH design) with a number of points equaling design A . LH designs are the standard designs used in the replication procedure to estimate first-order indices. The x-axis corresponds to the number of design points N and the y-axis to the value of the criterion.

FIGURE 5.1 – Log-log graph of averaged L_2 star discrepancy (left) and maximin (right) for designs M and A over 100 repetitions.



As expected, design M performs the best and the estimated slope of the discrepancy curve (-0.72) falls within the expected range for Sobol' sequences [MC94]. Although design A does not perform as well as design M , it outperforms a classic LH design for both criteria while allowing 15 refinement steps from 2^8 to 2^{12} points, instead of only 4 steps for design M .

Acknowledgements

This work is supported by :

- the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020),
- the United States National Science Foundation (grant DMS-1522687).

Titre : Reliable error estimation for Sobol' indices.

Auteurs : Laurent Gilquin^{a,b}, Lluís Antoni Jiménez Rugama^c.

a Inria Grenoble - Rhône-Alpes, Inovallee, 655 avenue de l'Europe, 38330 Montbonnot

b Univ. Grenoble Alpes, Jean Kuntzmann Laboratory, F-38000 Grenoble, France

CNRS, LJK, F-38000 Grenoble, France, Inria

c Illinois Institute of Technology, Rettaliata Engineering Center, 10 W 32st, Chicago, IL 60616

5.5 Introduction

Let f represent a deterministic numerical model in $[0, 1]^d$, $d \geq 1$. Sensitivity measures, also known as Sobol' indices, are used to assess which inputs of f are influential for the output. The normalized indices are scalars between 0 and 1 whose values are interpreted as follows : the closer to 1 the more influential the index. Alternatively, they can be interpreted as the percentage of the variance explained by the inputs. Among all Sobol' indices one can distinguish first-order and total effect indices. The first measure the effect of a single input, while the second measure the effect of a single input plus all its interactions with other inputs.

When dealing with complex numerical models, analytical expressions of Sobol' indices are often inaccessible. In such cases, one must rely on an estimation of these indices. The original estimation procedure is due to Sobol' [Sob93]. However, this procedure requires several model evaluations which are usually expensive. Later on, Saltelli [Sal02] proposed strategies to estimate sets of Sobol' indices at once through the use of a combinatorial formalisms. While elegant, these strategies still require a large number of model evaluations. A cost efficient alternative to estimate first-order indices was introduced in [MJ08]. This alternative, called the replication procedure, has recently been further studied in [TP15] and generalized to the estimation of closed second-order indices.

A practical problem concerning the use of these methods is how to quantify the number of model evaluations required to ensure that Sobol' estimates are accurate enough. This article addresses this challenge by proposing a reliable error bound for Sobol' indices based on digital sequences. The error bound is defined in terms of the discrete Walsh coefficients of the integrands involved in the Sobol' indices definition. We propose a sequential estimation procedure of Sobol' indices using the error bound as our stopping criterion. The procedure operates under the assumption that all integrands lie inside a particular cone of functions (see [HR16]).

Firstly, section 5.6 introduces Sobol' indices and reviews both Saltelli's strategy to estimate first-order and total effect Sobol' indices, and the replication procedure. Our main contribution is detailed in Section 5.7. There, we review the construction of

the error bound proposed in [HR16] for the estimation of integrals based on digital sequences, and then we generalize it for Sobol' indices. Section 5.8 is devoted to analyze the cost in terms of model evaluations of our sequential estimation algorithm. It combines the error bound in Section 5.7 and either one of the two estimation procedures of Section 5.6. We also discuss a potential improvement to estimate small first-order indices according to [Owe13a]. Finally, examples and illustrations of our procedure are presented in Section 5.9.

5.6 Background on Sobol' indices

5.6.1 Definition of Sobol' indices

Denote by $\mathbf{x} = (x_1, \dots, x_d)$ the vector of inputs of f . We assume that f is in some subset of $\mathbb{L}^2([0, 1]^d)$ for which $f(\mathbf{x})$ is defined for all $\mathbf{x} \in [0, 1]^d$, and $\mathcal{D} = \{1, \dots, d\}$ the set of dimension indexes. Let u be a subset of $\mathcal{D}$, $-u$ its complement and $|u|$ its cardinality. Then, $\mathbf{x}_u$ represents a point in $[0, 1]^{|u|}$ with components $x_j, j \in u$. Given two points $\mathbf{x}$ and $\mathbf{x}'$, the hybrid point $\mathbf{w} = (\mathbf{x}_u : \mathbf{x}'_{-u})$ is defined as $w_j = x_j$ if $j \in u$ and $w_j = x'_j$ if $j \notin u$.

The uncertainty on $\mathbf{x}$ is modeled by a uniform random vector, namely $\mathbf{x} \stackrel{\text{iid}}{\sim} \mathcal{U}([0, 1]^d)$. The Hoeffding decomposition [Hoe48, Sob93] of f is :

$$f(\mathbf{x}) = f_\emptyset + \sum_{u \subseteq \mathcal{D}, u \neq \emptyset} f_u(\mathbf{x}), \quad (5.6)$$

where :

$$\begin{aligned} f_\emptyset &= \mathbb{E}[f(\mathbf{x})] = \mu, \\ f_u(\mathbf{x}) &= \int_{[0, 1]^{|u|}} f(\mathbf{x}) d\mathbf{x}_{-u} - \sum_{v \subset u} f_v(\mathbf{x}). \end{aligned}$$

Due to orthogonality, the variance of equation (5.6) leads to the variance decomposition of f :

$$\sigma^2 = \text{Var}[f(\mathbf{x})] = \sum_{u \subseteq \mathcal{D}, u \neq \emptyset} \sigma_u^2, \quad \text{with} \quad \sigma_u^2 = \int_{[0, 1]^{|u|}} f_u(\mathbf{x})^2 d\mathbf{x}_u.$$

From this decomposition, one can express the following two quantities :

$$\underline{\tau}_u^2 = \sum_{v \subseteq u} \sigma_v^2, \quad \bar{\tau}_u^2 = \sum_{v \cap u \neq \emptyset} \sigma_v^2, \quad u \subsetneq \mathcal{D}.$$

These two quantities $\underline{\tau}_u^2$ and $\bar{\tau}_u^2$ measure the importance of variables $\mathbf{x}_u$: $\underline{\tau}_u^2$ quantifies the main effect of $\mathbf{x}_u$, that is the effect of all interactions between variables in $\mathbf{x}_u$, and $\bar{\tau}_u^2$ quantifies the main effect of $\mathbf{x}_u$ plus the effect of all interactions between variables in $\mathbf{x}_u$ and variables in $\mathbf{x}_{-u}$.

Both $\underline{\tau}_u^2$ and $\bar{\tau}_u^2$ satisfy the following relations : $0 \leq \underline{\tau}_u^2 \leq \bar{\tau}_u^2$ and $\underline{\tau}_u^2 = \sigma^2 - \bar{\tau}_{-u}^2$. These two measures are commonly found in the literature in their normalized form : $\underline{S}_u = \underline{\tau}_u^2/\sigma^2$ is the closed $|u|$ -order Sobol' index for inputs u , while $\bar{S}_u = \bar{\tau}_u^2/\sigma^2$ is the total effect Sobol' index of order $|u|$.

In our framework, we are only interested in single input indices, namely $|u| = 1$. The computation of the normalized indices is performed based on the following integral formulas for their numerators :

$$\underline{\tau}_u^2 = \int_{[0,1]^{2d}} (f(\mathbf{x}) - f(\mathbf{x}_u : \mathbf{x}'_{-u})) f(\mathbf{x}') d\mathbf{x} d\mathbf{x}', \quad (5.7)$$

$$\bar{\tau}_u^2 = \frac{1}{2} \int_{[0,1]^{d+1}} (f(\mathbf{x}') - f(\mathbf{x}_u : \mathbf{x}'_{-u}))^2 d\mathbf{x}_u d\mathbf{x}', \quad u \in \mathcal{D}, \quad (5.8)$$

while variance and mean of f are evaluated as :

$$\begin{aligned} \sigma^2 &= \int_{[0,1]^d} f(\mathbf{x})^2 d\mathbf{x} - \mu^2, \\ \mu &= \int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x}, \end{aligned} \quad (5.9)$$

Usually the complexity of f causes the solution of integrals (5.7), (5.8) and (5.9) to be intractable. In such cases, one can instead estimate these quantities.

5.6.2 Estimation of Sobol' indices

In this section we review two Monte Carlo procedures for the estimation of Sobol' indices. A design is a point set $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$ where each point is obtained by sampling each variable x_j n times. Each row of the design is a point $\mathbf{x}_i$ in $[0,1]^d$ and each column of the design refers to a variable x_j . Consider $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$ and $\mathcal{P}' = \{\mathbf{x}'_i\}_{i=0}^{n-1}$ two designs where $(\mathbf{x}_i, \mathbf{x}'_i) \stackrel{\text{iid}}{\sim} \mathcal{U}([0,1]^{2d})$. One way to estimate the two quantities (5.7) and (5.8) is via :

$$\widehat{\underline{\tau}}_u^2 = \frac{1}{n} \sum_{i=0}^{n-1} (f(\mathbf{x}_i) - f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u})) f(\mathbf{x}'_i), \quad (5.10)$$

$$\widehat{\bar{\tau}}_u^2 = \frac{1}{2n} \sum_{i=0}^{n-1} (f(\mathbf{x}'_i) - f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u}))^2, \quad u \in \mathcal{D}, \quad (5.11)$$

using for σ^2 :

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i)^2 - \hat{\mu}^2, \quad \text{with} \quad \hat{\mu} = \frac{1}{n} \sum_{i=0}^{n-1} f(\mathbf{x}_i). \quad (5.12)$$

Then, the Sobol' indices estimators are :

$$\widehat{\underline{S}}_u = \widehat{\underline{\tau}}_u^2 / \hat{\sigma}^2, \quad \widehat{\bar{S}}_u = \widehat{\bar{\tau}}_u^2 / \hat{\sigma}^2. \quad (5.13)$$

Using this method, the estimation of a single pair $(\widehat{S}_u, \widehat{S}_u)$ requires $3n$ evaluations of the model f . Thus, for all first order and total effect indices, one would need $3nd$ model evaluations. Using a combinatorial formalism, in [Sal02] Saltelli proposes the following estimation strategy :

Théorème 5.1. *The $d+2$ designs $\{\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u}\}_{i=0}^{n-1}$ constructed for $u \in \{\emptyset, \{1\}, \dots, \{d\}, \mathcal{D}\}$ allow to estimate all first-order and total effect Sobol' indices at a cost of $n(d+2)$ evaluations of the model.*

The main idea of the theorem is that there is no need to reevaluate $f(\mathbf{x}_i)$ and $f(\mathbf{x}'_i)$ for each u . Hence, we can simply evaluate $f(\mathbf{x}_i)$ n times, $f(\mathbf{x}'_i)$ n times, and $f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u})$ nd times, which accounts for the $n(d+2)$ evaluations.

The $d+2$ designs of Theorem 5.1 are obtained by substituting columns of $\mathcal{P}$ for columns of $\mathcal{P}'$ accordingly to u . While elegant, this approach requires a number of model evaluations that grows linearly with respect to the input space dimension.

A more efficient alternative to evaluate all first-order indices was proposed in [MJ08] and only requires $2n$ model evaluations. This alternative relies on the construction of two replicated designs. The notion of replicated designs was first introduced by McKay through its replicated Latin Hypercubes in [McK95]. In order to apply this definition to other types of points, in [GRA⁺16] we generalized this notion to the following :

Définition 5.1. *Let $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$ and $\mathcal{P}' = \{\mathbf{x}'_i\}_{i=0}^{n-1}$ be two point sets in $[0, 1]^d$. Let $\mathcal{P}^u = \{\mathbf{x}_{i,u}\}_{i=0}^{n-1}$ (resp. $\mathcal{P}'^u$), $u \subsetneq \mathcal{D}$, denote the subset of dimensions of $\mathcal{P}$ (resp. $\mathcal{P}'$) indexed by u . We say that $\mathcal{P}$ and $\mathcal{P}'$ are two replicated designs of order $a \in \{1, \dots, d-1\}$ if for any $u \subsetneq \mathcal{D}$ such that $|u| = a$, $\mathcal{P}^u$ and $\mathcal{P}'^u$ are the same point set in $[0, 1]^a$. We define by π_u the permutation that rearranges the rows of $\mathcal{P}'^u$ into $\mathcal{P}^u$.*

The method introduced in [MJ08] allows to estimate all first-order Sobol' indices with only two replicated designs of order 1. The key point of this method is to use the permutations resulting from the structure of the two replicated designs to mimic the hybrid points in equation (5.10).

More precisely, let $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{n-1}$ and $\mathcal{P}' = \{\mathbf{x}'_i\}_{i=0}^{n-1}$ be two replicated designs of order 1. Denote by $\{y_i\}_{i=0}^{n-1} = \{f(\mathbf{x}_i)\}_{i=0}^{n-1}$ and $\{y'_i\}_{i=0}^{n-1} = \{f(\mathbf{x}'_i)\}_{i=0}^{n-1}$ the two sets of model evaluations obtained with $\mathcal{P}$ and $\mathcal{P}'$. From Definition 5.1, we know that $\mathbf{x}'_{\pi_u(i),u} = \mathbf{x}_{i,u}$. Then,

$$\begin{aligned} y'_{\pi_u(i)} &= f(\mathbf{x}'_{\pi_u(i),u} : \mathbf{x}'_{\pi_u(i),-u}) \\ &= f(\mathbf{x}_{i,u} : \mathbf{x}'_{\pi_u(i),-u}). \end{aligned}$$

Hence, each $\underline{\tau}_u^2$ can be estimated via formula (5.10) by using $y'_{\pi_u(i)}$ instead of $f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u})$ without requiring further model evaluations for each u . This estimation method has been deeply studied and generalized in Tissot et al. [TP15] to the case of closed second-order indices. In the following we will refer to this method as the replication procedure.

5.6.3 Towards a reliable estimation

The aim of this article is to propose a sequential procedure to estimate first-order and total effect Sobol' indices. A practical problem concerning the estimation of these indices is how large to choose the number of evaluations to ensure that Sobol' estimates are accurate enough. Asymptotical results show that Sobol' estimates are normally distributed ([JKL⁺14, Proposition 2.2], [TP15, Proposition 3.5]). As a consequence, errors can be estimated through confidence intervals. However, these error estimates are only guaranteed asymptotically as the number of model evaluations goes to infinity.

Additional sequential procedures are the replicated procedure and McKay's procedure respectively proposed in [GACP16] and [Ton10]. Nevertheless, in those two cases, the stopping criterion is a quantity of interest purely empirical, built directly upon the estimates. Such stopping criteria often involve hyper-parameters difficult to tweak but more importantly, fail to guarantee any error bound on the estimates.

Our sequential procedure stands apart from others since it proposes a robust stopping criterion, not costly to compute. This criterion is an error bound based on the Walsh series decomposition of the integrands in (5.7), (5.8) and (5.9), and exploits the group properties of digital nets. As such, our procedure relies on an iterative construction of Sobol' sequences. This construction is performed accordingly to the multiplicative approach presented in [GRA⁺16].

The description of the error bound is introduced in the following section and our sequential procedure is detailed in section 5.8.

5.7 Reliable error bound for Sobol' indices

We start by reviewing the construction of the error bound proposed in [HR16] for the estimation of d -dimensional integrals. Then, we present an extension of this error bound for normalized Sobol' indices. This extension is built upon the integral formula of a Sobol' index.

5.7.1 Reliable integral estimation using digital sequences

We assume we have an embedded sequence of digital nets in base b as in [HR16, Sec. 2-3],

$$\mathcal{P}_0 = \{\mathbf{0}\} \subset \dots \subset \mathcal{P}_m = \{\mathbf{x}_i\}_{i=0}^{b^m-1} \subset \dots \subset \mathcal{P}_\infty = \{\mathbf{x}_i\}_{i=0}^\infty.$$

Each $\mathcal{P}_m$ has a group structure under the digitwise addition :

$$\mathbf{x} \oplus \mathbf{t} = \left(\sum_{\ell=1}^{\infty} [(x_{j\ell} + t_{j\ell}) \bmod b] b^{-\ell} \pmod{1} \right)_{j=1}^d,$$

where $x_{j\ell}$ and $t_{j\ell}$ are the b -adic decompositions of the j^{th} component of points $\mathbf{x}$ and $\mathbf{t}$.

To relate the group structure of $\mathcal{P}_m$ with the integration error, we introduce the *dual net* which establishes the relationship between any digital net and the *wavenumber* space of non-negative integers $\mathbb{N}_0^d$. A dual net is

$$\begin{aligned}\mathcal{P}_m^\perp &= \{\mathbf{k} \in \mathbb{N}_0^d : \langle \mathbf{k}, \mathbf{x} \rangle = 0, \mathbf{x} \in \mathcal{P}_m\}, \\ \langle \mathbf{k}, \mathbf{x} \rangle &= \sum_{j=1}^d \sum_{\ell=0}^{\infty} k_{j\ell} x_{j,\ell+1} \pmod{b},\end{aligned}$$

and inherits the same embedded structure as for digital nets,

$$\mathcal{P}_0^\perp = \mathbb{N}_0^d \supset \cdots \supset \mathcal{P}_\infty^\perp = \{\mathbf{0}\}. \quad (5.14)$$

As shown in [HR16, Sec. 3], the group structure of digital nets guarantees the property below affecting any Walsh basis $\varphi_{\mathbf{k}}(\mathbf{x}) = e^{2\pi\sqrt{-1}\langle \mathbf{k}, \mathbf{x} \rangle/b}$,

$$\frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} \varphi_{\mathbf{k}}(\mathbf{x}) = \begin{cases} 1, & \mathbf{k} \in \mathcal{P}_m^\perp, \\ 0, & \mathbf{k} \notin \mathcal{P}_m^\perp. \end{cases} \quad (5.15)$$

Therefore, considering the Walsh decomposition of any $f \in \mathbb{L}^2([0, 1]^d)$:

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}_{\mathbf{k}} \varphi_{\mathbf{k}}(\mathbf{x}),$$

and,

$$I = \int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x},$$

property (5.15) leads to

$$\begin{aligned} \left| I - \frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} f(\mathbf{x}) \right| &= \left| \hat{f}_{\mathbf{0}} - \frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} \sum_{\mathbf{k} \in \mathbb{N}_0^d} \hat{f}_{\mathbf{k}} \varphi_{\mathbf{k}}(\mathbf{x}) \right|, \\ &= \left| \sum_{\mathbf{k} \in \mathcal{P}_m^\perp \setminus \{\mathbf{0}\}} \hat{f}_{\mathbf{k}} \right| \\ &\leq \sum_{\mathbf{k} \in \mathcal{P}_m^\perp \setminus \{\mathbf{0}\}} |\hat{f}_{\mathbf{k}}|. \end{aligned} \quad (5.16)$$

Based on the size of $|\hat{f}_{\mathbf{k}}|$ and the structure of dual nets (5.14), in [HR16, Sec. 4.1] we proposed an ordering of the *wavenumbers* $\mathbf{k}(\cdot) : \mathbb{N}_0 \rightarrow \mathbb{N}_0^d$. Defining $\hat{f}_\kappa = \hat{f}_{\mathbf{k}(\kappa)}$, the error bound (5.16) becomes,

$$\left| I - \frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} f(\mathbf{x}) \right| \leq \sum_{\lambda=1}^{\infty} |\hat{f}_{\lambda b^m}|. \quad (5.17)$$

Nonetheless, because the knowledge of the Walsh coefficients $\hat{f}_\kappa$ is not assumed, we will estimate them using the fast transform obtained with the precomputed function values, and refer to them as $\tilde{f}_{m,\kappa}$,

$$\tilde{f}_{m,\kappa} = \tilde{f}_{m,\mathbf{k}(\kappa)} = \frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} f(\mathbf{x}) \overline{\varphi_{\mathbf{k}(\kappa)}(\mathbf{x})}.$$

Note that for function values evaluated at $\mathcal{P}_m$, one only generates b^m discrete Walsh coefficients $\tilde{f}_{m,\kappa}$.

For $\ell, m \in \mathbb{N}_0$ and $\ell \leq m$ we introduce the following notation,

$$\begin{aligned} S_m(f) &= \sum_{\kappa=[b^{m-1}]}^{b^m-1} |\hat{f}_\kappa|, \quad \hat{S}_{\ell,m}(f) = \sum_{\kappa=[b^{\ell-1}]}^{b^\ell-1} \sum_{\lambda=1}^{\infty} |\hat{f}_{\kappa+\lambda b^m}|, \\ \check{S}_m(f) &= \hat{S}_{0,m}(f) + \cdots + \hat{S}_{m,m}(f) = \sum_{\kappa=b^m}^{\infty} |\hat{f}_\kappa|, \\ \check{S}_{\ell,m}(f) &= \sum_{\kappa=[b^{\ell-1}]}^{b^\ell-1} |\tilde{f}_{m,\kappa}|. \end{aligned}$$

Finally, we define the set of functions $\mathcal{C}$,

$$\begin{aligned} \mathcal{C} := \{f \in \mathbb{L}^2([0,1]^d) : \hat{S}_{\ell,m}(f) \leq \hat{\omega}(m-\ell) \check{S}_m(f), \quad \ell \leq m, \\ \check{S}_m(f) \leq \hat{\omega}(m-\ell) S_\ell(f), \quad \ell_* \leq \ell \leq m\}. \end{aligned} \quad (5.18)$$

for $\ell_* \in \mathbb{N}$, $\hat{\omega}$ and $\hat{\omega}$ two non-negative valued functions with $\lim_{m \rightarrow \infty} \hat{\omega}(m) = 0$, and such that $\hat{\omega}(r) \hat{\omega}(r) < 1$ for some fixed $r \in \mathbb{N}$.

In [HR16, Sec. 4.2] it is shown that for any $f \in \mathcal{C}$,

$$\left| I - \frac{1}{b^m} \sum_{\mathbf{x} \in \mathcal{P}_m} f(\mathbf{x}) \right| \leq \check{S}_{\ell,m}(f) \underbrace{\frac{\hat{\omega}(m) \hat{\omega}(m-\ell)}{1 - \hat{\omega}(m-\ell) \hat{\omega}(m-\ell)}}_{\mathfrak{C}(m)} = \varepsilon_{\hat{I}}, \quad (5.19)$$

where one may increase m until the error bound ε_I is small enough.

Details concerning the algorithm, the mapping of the *wavenumber* space, or the meaning and properties of $\mathcal{C}$, are provided in [HR16].

For our problem, only Sobol' sequences [Sob67] have been considered. Their major interest comes from their fast and easy implementation. Further details concerning Sobol' sequences can be found in [Lem09, Nie92a].

5.7.2 Extension to Sobol' indices

The idea here is to extend the definition of the error bound (5.19) to Sobol' indices. To do so, we consider the two integral formulas of the first-order and total

effect Sobol' indices with $|u| = 1$:

$$\begin{aligned}\underline{S}_u(\mathbf{I}) &= \frac{\int_{[0,1]^{2d}} (f(\mathbf{x}) - f(\mathbf{x}_u : \mathbf{x}'_{-u})) f(\mathbf{x}') d\mathbf{x} d\mathbf{x}'}{\int_{[0,1]^d} f(\mathbf{x})^2 d\mathbf{x} - \left(\int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x} \right)^2}, \\ \underline{S}_u(\mathbf{I}) &= \frac{I_1}{I_3 - (I_4)^2}, \\ \overline{S}_u(\mathbf{I}) &= \frac{\frac{1}{2} \int_{[0,1]^{d+1}} (f(\mathbf{x}') - f(\mathbf{x}_u : \mathbf{x}'_{-u}))^2 d\mathbf{x}_u d\mathbf{x}'}{\int_{[0,1]^d} f(\mathbf{x})^2 d\mathbf{x} - \left(\int_{[0,1]^d} f(\mathbf{x}) d\mathbf{x} \right)^2}, \\ \overline{S}_u(\mathbf{I}) &= \frac{I_2}{I_3 - (I_4)^2},\end{aligned}$$

where $\mathbf{I} = (I_1, I_2, I_3, I_4)$ is a vector of integral values. $\underline{S}_u(\mathbf{I})$ and $\overline{S}_u(\mathbf{I})$ are defined as functions over the vector $\mathbf{I}$. If we estimate $\mathbf{I}$ by $\hat{\mathbf{I}}$ with vector of error bounds $\varepsilon_{\hat{\mathbf{I}}} = (\varepsilon_{\hat{I}_1}, \varepsilon_{\hat{I}_2}, \varepsilon_{\hat{I}_3}, \varepsilon_{\hat{I}_4})$ according to (5.19), we know that $\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})} = [\hat{\mathbf{I}} - \varepsilon_{\hat{\mathbf{I}}}, \hat{\mathbf{I}} + \varepsilon_{\hat{\mathbf{I}}}]$.

Then, as an alternative to the common Sobol' indices estimators (5.13), we can define the following two estimators with their respective error bounds :

$$\begin{aligned}\hat{\underline{S}}_u &= \frac{1}{2} \left(\min \left(\max_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \underline{S}_u(\mathbf{I}), 1 \right) + \max \left(\min_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \underline{S}_u(\mathbf{I}), 0 \right) \right) \\ \varepsilon_{\hat{\underline{S}}_u} &= \frac{1}{2} \left(\min \left(\max_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \underline{S}_u(\mathbf{I}), 1 \right) - \max \left(\min_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \underline{S}_u(\mathbf{I}), 0 \right) \right)\end{aligned}\tag{5.20}$$

and,

$$\begin{aligned}\hat{\overline{S}}_u &= \frac{1}{2} \left(\min \left(\max_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \overline{S}_u(\mathbf{I}), 1 \right) + \max \left(\min_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \overline{S}_u(\mathbf{I}), 0 \right) \right) \\ \varepsilon_{\hat{\overline{S}}_u} &= \frac{1}{2} \left(\min \left(\max_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \overline{S}_u(\mathbf{I}), 1 \right) - \max \left(\min_{\mathbf{I} \in B_{\varepsilon_{\hat{\mathbf{I}}}(\hat{\mathbf{I}})}} \overline{S}_u(\mathbf{I}), 0 \right) \right)\end{aligned}\tag{5.21}$$

Because numerator and denominator are both known to be positive, maximizing $\underline{S}_u(\mathbf{I})$ (resp. $\overline{S}_u(\mathbf{I})$) is done through maximizing the numerator I_1 (resp. I_2) and minimizing the denominator $I_3 - I_4^2$. Analogously, to minimize $\underline{S}_u(\mathbf{I})$ (resp. $\overline{S}_u(\mathbf{I})$) one minimizes the numerator I_1 (resp. I_2) and maximizes the denominator $I_3 - I_4^2$.

Under the assumption that each integrand in $\underline{S}_u$ and $\overline{S}_u$ lies inside $\mathcal{C}$, these new estimators satisfy :

$$\underline{S}_u \in \left[\hat{\underline{S}}_u - \varepsilon_{\hat{\underline{S}}_u}, \hat{\underline{S}}_u + \varepsilon_{\hat{\underline{S}}_u} \right], \quad \overline{S}_u \in \left[\hat{\overline{S}}_u - \varepsilon_{\hat{\overline{S}}_u}, \hat{\overline{S}}_u + \varepsilon_{\hat{\overline{S}}_u} \right].$$

For the rest of the paper, we will consider $\hat{\underline{S}}_u$ and $\hat{\overline{S}}_u$ as defined in formulas (5.20) and (5.21).

5.8 Sequential estimation procedure

The sequential estimation procedure we propose combines error bounds $\varepsilon_{\hat{S}_u}$ and $\varepsilon_{\hat{\bar{S}}_u}$ presented in the previous section with either one of the two estimation strategies of section 5.6.2 : Saltelli's strategy and the replication procedure.

We start by detailing our procedure under the form of an algorithm. Then, we discuss a possible improvement by considering a new estimator introduced in [Owe13a] for the estimation of first-order indices.

5.8.1 Sequential algorithm and cost

Algorithm 5.1 summarizes the main steps of our sequential procedure. First, one must fix the tolerance $\varepsilon > 0$ at which Sobol' indices must be estimated, and set $m = \ell_* + r$, where ℓ_* and r are the two hyper-parameters defined in (5.18). Then, one must construct the two designs $\mathcal{P}_m = \{\mathbf{x}_i\}_{i=0}^{2^m-1}$ and $\mathcal{P}'_m = \{\mathbf{x}'_i\}_{i=0}^{2^m-1}$ at Step 5 according to the multiplicative approach detailed in [GRA⁺16]. The choice of ℓ_* must be large enough to ensure that each integrand of $\underline{S}_u$ and $\bar{S}_u$ (Section 5.7.2) is in $\mathcal{C}$ (see (5.18)).

Under this construction, $\mathcal{P}_m$ and $\mathcal{P}'_m$ correspond to the first 2^m points of two independent Sobol' sequences. Furthermore, they possess the structure of two replicated designs of order 1, as shown in [GRA⁺16]. $\mathcal{P}_m$ and $\mathcal{P}'_m$ can also be scrambled as long as both point sets share the same scrambling dimensionwise.

$\mathcal{P}_m$ and $\mathcal{P}'_m$ can be used with Saltelli's strategy to estimate all first-order indices and total effect Sobol' indices. This option is referred as Variant *A* in Algorithm 5.1.

Alternatively, $\mathcal{P}_m$ and $\mathcal{P}'_m$ can be used with the replication procedure to estimate all first-order Sobol' indices. This option is referred as Variant *B* in Algorithm 5.1.

In both cases we always check if the respective error bounds $\varepsilon_{\hat{\underline{S}}_u}$ and $\varepsilon_{\hat{\bar{S}}_u}$ are lower than the tolerance ε . For Variant *A* the stopping criterion is $\varepsilon_{\hat{\underline{S}}_u} \leq \varepsilon$ and $\varepsilon_{\hat{\bar{S}}_u} \leq \varepsilon$ for all $u \in \mathcal{D}$. For Variant *B* it is $\varepsilon_{\hat{\underline{S}}_u} \leq \varepsilon$ for all $u \in \mathcal{D}$.

If the stopping criterion is satisfied, the algorithm stops and Sobol' estimates are returned. Otherwise, m is incremented by one to perform a new estimation.

The cost of our algorithm, in terms of model evaluations, varies whether Variant *A* or Variant *B* is selected. To discuss this cost we note by m^* the ending iteration. If Variant *A* is selected, the cost of our algorithm is :

$$\sum_{u \in \mathcal{D}} 2^{m_u} + 2 \times 2^{m^*}, \quad m^* = \max_{u \in \mathcal{D}} m_u,$$

where :

- 2^{m_u} is the number of evaluations $f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u})$ used to estimate both the first-order index $\underline{S}_u$ and the total effect index $\bar{S}_u$,
- $2 \times 2^{m^*}$ is the number of evaluations $f(\mathbf{x}_i)$ and $f(\mathbf{x}'_i)$ used in the estimation of each first-order and total effect indices.

Algorithm 5.1 Sequential estimation of Sobol' indices

```

1: choose  $\varepsilon > 0$ 
2: set :  $m \leftarrow \ell_* + r$ 
3:  $bool \leftarrow false$ 
4: while ! $bool$  do
5:    $\mathcal{P}_m \leftarrow \mathcal{P}_{m-1} \cup B_m$ 
    $\mathcal{P}'_m \leftarrow \mathcal{P}'_{m-1} \cup B'_m$ 
6:   for  $u = 1, \dots, d$  do
7:     if Variant  $A$  then
8:       if ! $bool_u$  then
9:         compute  $\hat{\underline{S}}_u$  and  $\hat{\bar{S}}_u$  with formulas (5.20) and (5.21), and Saltelli's
           strategy
10:         $bool_u \leftarrow \varepsilon_{\hat{\underline{S}}_u} \leq \varepsilon \ \& \ \varepsilon_{\hat{\bar{S}}_u} \leq \varepsilon$ 
11:         $m_u \leftarrow m$ 
12:      if Variant  $B$  then
13:        if ! $bool_u$  then
14:          estimate  $\hat{\underline{S}}_u$  with formula (5.20) and the replication procedure
15:           $bool_u \leftarrow \varepsilon_{\hat{\underline{S}}_u} \leq \varepsilon$ 
16:           $m_u \leftarrow m$ 
17:       $bool \leftarrow \forall u : bool_u$ 
18:       $m \leftarrow m + 1$ 
19: return the Sobol' estimates.

```

If all m_u are equal, the cost of Variant A becomes $2^{m^*}(d+2)$ and we recover the cost specified in Theorem 5.1 with $n = 2^{m^*}$.

If Variant B is selected, the cost of our algorithm equals $2 \times 2^{m^*}$. This cost corresponds to the one of the replication procedure introduced in Section 5.6.2, where $2n = 2 \times 2^{m^*}$ independent of d .

5.8.2 Improvement

When $\underline{S}_u$ is small, it usually becomes harder to estimate. For this reason, we consider the use of a new estimator to evaluate small first-order Sobol' indices in Variant A . This estimator called "Correlation 2" has been introduced by Owen in [Owe13a]. In this article, he discussed and highlighted the efficiency of "Correlation 2" when estimating small first-order indices. Our aim is to show that the use of "Correlation 2" in Variant A may reduce the total number of model evaluations. This new estimator is,

$$\widehat{\tau}_{u,2}^2 = \frac{1}{n} \sum_{i=0}^{n-1} (f(\mathbf{x}_i) - f(\mathbf{z}_{i,u} : \mathbf{x}_{i,-u}))(f(\mathbf{x}_{i,u} : \mathbf{x}'_{i,-u}) - f(\mathbf{x}'_i)),$$

where $(\mathbf{x}_i, \mathbf{x}'_i, \mathbf{z}_i) \stackrel{\text{iid}}{\sim} \mathcal{U}([0, 1]^{3d})$. It requires an additional set of n model evaluations to estimate $\underline{\tau}_u^2$.

We discuss below the potential improvement brought by using “Correlation 2” in Variant *A*. The idea is to replace the current estimator (5.10) by “Correlation 2” for each small first-order index.

Assume that the number of small first-order indices is known and equals γ . We denote by $u_1, \dots, u_\gamma$ the indices of the corresponding inputs and $\Gamma = \{1, \dots, \gamma\}$. The cost of Variant *A* including “Correlation 2” becomes,

$$\sum_{j \in \Gamma} 2^{m''_{u_j}} + \sum_{j \in \Gamma} 2^{m'_{u_j}} + \sum_{j \in \mathcal{D} \setminus \Gamma} 2^{m_{u_j}} + 2 \times 2^{m^*}, \quad (5.22)$$

where :

- for $j \in \Gamma$, $2^{m''_{u_j}}$ is the number of evaluations $f(\mathbf{z}_{i,u_j} : \mathbf{x}_{i,-u_j})$ to estimate $\underline{S}_{u_j}$,
- for $j \in \Gamma$, $2^{m'_{u_j}}$ is the number of evaluations $f(\mathbf{x}_{i,u_j} : \mathbf{x}'_{i,-u_j})$ to estimate both $\underline{S}_{u_j}$ and $\bar{S}_{u_j}$,
- likewise, for $j \in \mathcal{D} \setminus \Gamma$, $2^{m_{u_j}}$ is the number of evaluations $f(\mathbf{x}_{i,u_j} : \mathbf{x}'_{i,-u_j})$ to estimate both $\underline{S}_{u_j}$ and $\bar{S}_{u_j}$,
- $2 \times 2^{m^*}$ is the number of evaluations $f(\mathbf{x}_i)$ and $f(\mathbf{x}'_i)$ used in the estimation of each first-order and total effect index.

We recall the cost of Variant *A* without “Correlation 2”,

$$\sum_{j \in \Gamma} 2^{m_{u_j}} + \sum_{j \in \mathcal{D} \setminus \Gamma} 2^{m_{u_j}} + 2 \times 2^{m^*}. \quad (5.23)$$

The difference (5.22) – (5.23) equals :

$$\sum_{j \in \Gamma} 2^{m_{u_j}} \left(2^{m''_{u_j} - m_{u_j}} + 2^{m'_{u_j} - m_{u_j}} - 1 \right) = \sum_{j \in \Gamma} c_j. \quad (5.24)$$

Hence, the sign of this difference indicates whether or not using “Correlation 2” brings an improvement to Variant *A*. We distinguish two cases :

- 1) for $j \in \Gamma$, if the total effect index $\bar{S}_{u_j}$ requires as much or more evaluations than the first-order index $\underline{S}_{u_j}$. Since the total effect estimator is the same, as a consequence we have $m'_{u_j} = m_{u_j}$ and $c_j > 0$.
- 2) for $j \in \Gamma$, if the total effect index $\bar{S}_{u_j}$ requires less evaluations than the first-order index $\underline{S}_{u_j}$. In this case, if both $m''_{u_j} < m_{u_j}$ and $m'_{u_j} < m_{u_j}$ then $c_j \leq 0$.

Overall we expect to observe case 2) more often than case 1). Indeed, the numerator of $\underline{S}_u$ requires to estimate $2d$ dimensional integrals against only $d + 1$ dimensional integrals for the numerator of $\bar{S}_u$. Hence, it seems reasonable to expect that it will take less points to estimate $\bar{S}_u$ than $\underline{S}_u$.

Furthermore, in case 2), we expect the two conditions $m''_{u_j} < m_{u_j}$ and $m'_{u_j} < m_{u_j}$ to usually hold as in [Owe13a], “Correlation 2” is shown to perform better for small

first-order indices. To support this discussion, numerical examples are presented in Section 5.9.

In practice, one does not know which are the small Sobol' indices $(u_1, \dots, u_\gamma)$. To overcome this issue, we propose the following alternative for Variant A. If at the end of the first iteration $\widehat{S}_u$ is smaller than a threshold (in our case 0.1, as suggested by Owen), then the estimator (5.10) is switched to "Correlation 2" for this particular u and a third Sobol' sequence $\mathcal{P}''_m = \{z_i\}_{i=0}^{2^m-1}$ is constructed to obtain the corresponding evaluations $f(z_{i,u} : x_{i,-u})$.

5.9 Applications

We illustrate our sequential estimation procedure with two classical test functions and one real example. In each case, Sobol' indices are estimated with the following three variants :

- Variant A.a as Variant A without using "Correlation 2".
- Variant A.b as Variant A using "Correlation 2".
- Variant B.

The threshold to decide whether or not a first-order index is small is set to 0.1.

For the two test functions, results of the three variants are compared based on the true estimation errors. These errors correspond to the absolute difference between the true values and their estimates :

$$\delta_{\underline{S}_u} = \left| \underline{S}_u - \widehat{\underline{S}}_u \right|, \quad \delta_{\overline{S}_u} = \left| \overline{S}_u - \widehat{\overline{S}}_u \right|.$$

We also compare the total number of evaluations of each variant. Results are averaged over 100 repetitions using Owen's scrambled Sobol' sequences [HH03, Owe95]. For each repetition, we fix the tolerance $\varepsilon = 5.10^{-3}$, set $\ell_* = 5$, and $r = 4$ (the algorithm starts with 512 Sobol' points). The algorithm error bound factor defined in (5.19) is set to $\mathfrak{C}(m) = 10 \times 2^{-m}$.

The failure rates $r_{\delta_{\underline{S}_u}}$ and $r_{\delta_{\overline{S}_u}}$ of each estimate are evaluated over the 100 repetitions. These rates are the proportion of the repetitions in which the true estimation errors failed to satisfy $\delta_{\underline{S}_u} \leq \varepsilon$ and $\delta_{\overline{S}_u} \leq \varepsilon$. When $\delta_{\underline{S}_u} > \varepsilon$ and $\delta_{\overline{S}_u} > \varepsilon$, we can infer that our integration parameters are not conservative enough and the integrands fell outside the cone (5.19).

5.9.1 Classical test functions

The two classical test functions considered in this article are the g-function introduced by Sobol' [Sob93], and the function introduced by Bratley *et al.* [BFN92]. The idea is to test our method over two categories of functions : additive (Bratley *et al.* function) and multiplicative (g-function).

5.9.1.1 Sobol' g-function

The g-function is defined as follows :

$$f(\mathbf{x}) = \prod_{j=1}^d g_j(x_j), \quad g_j(x_j) = \frac{|4x_j - 2| + a_j}{1 + a_j}, \quad a_j \geq 0.$$

Each value a_j determines the relative importance of the x_j . When the value of a_j gets closer to zero the variable x_j becomes more influent. For this example, we chose $d = 6$ and $a_1 = 0, a_2 = 0.5, a_3 = 3, a_4 = 9, a_5 = 99$, and $a_6 = 99$.

Table 5.1 shows the averaged estimation errors obtained with Variant *A.a* as well as the averaged total number of evaluations performed. Table 5.2 shows the same results obtained with Variant *A.b*.

TABLE 5.1 – Averaged estimation errors $\delta_{\underline{S}_u}$, $\delta_{\overline{S}_u}$ and total number of evaluations for Variant *A.a*.

input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r_{\delta_{\underline{S}_u}}$	$\overline{S}_u$	$\delta_{\overline{S}_u}$	$r_{\delta_{\overline{S}_u}}$
x_1	0.5868	0.0004	0.02	0.6901	0.0005	0.02
x_2	0.2608	0.0006	0.02	0.3562	0.0004	0
x_3	0.0367	0.0011	0.03	0.0563	0.0008	0
x_4	0.0058	0.0012	0.02	0.0092	0.0003	0
x_5	5.10^{-5}	0.0003	0	9.10^{-5}	$< 10^{-4}$	0
x_6	5.10^{-5}	0.0003	0	9.10^{-5}	$< 10^{-4}$	0
Total number of evaluations : 63 088						

TABLE 5.2 – Averaged estimation errors $\delta_{\underline{S}_u}$, $\delta_{\overline{S}_u}$ and total number of evaluations for Variant *A.b*.

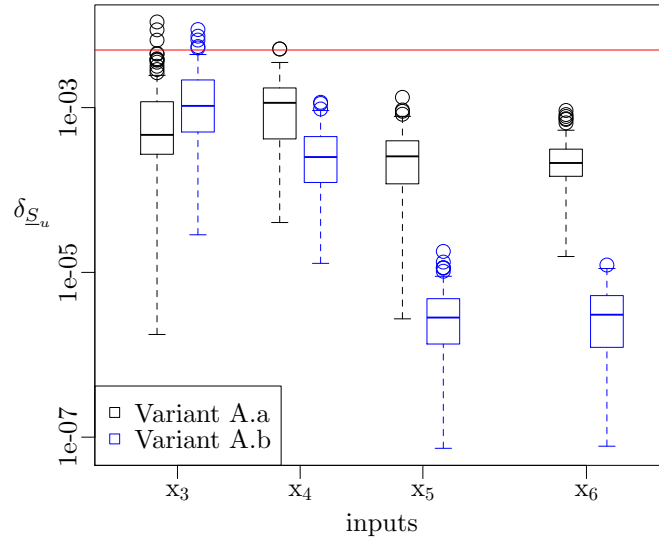
input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r_{\delta_{\underline{S}_u}}$	$\overline{S}_u$	$\delta_{\overline{S}_u}$	$r_{\delta_{\overline{S}_u}}$
x_1	0.5868	0.0005	0.03	0.6901	0.0006	0.02
x_2	0.2608	0.0007	0.03	0.3562	0.0003	0
x_3	0.0367	0.0015	0.05	0.0563	0.0009	0
x_4	0.0058	0.0003	0	0.0092	0.0003	0
x_5	5.10^{-5}	$< 10^{-4}$	0	9.10^{-5}	$< 10^{-4}$	0
x_6	5.10^{-5}	$< 10^{-4}$	0	9.10^{-5}	$< 10^{-4}$	0
Total number of evaluations : 62 703						

The main observation is that both approaches give similar results both in terms

of estimation errors, failure rates and total number of evaluations. The use of “Correlation 2” in Variant *A.b* to estimate the four small first-order Sobol' indices $\underline{S}_3, \underline{S}_4, \underline{S}_5, \underline{S}_6$ does not seem to bring much improvement. We propose to draw boxplots of the estimation errors to further investigate the results.

Figure 5.2 shows boxplots of the 100 estimation errors $\delta_{\underline{S}_u}$ obtained with both Variant *A.a* and Variant *A.b* for the four inputs x_3, x_4, x_5, x_6 . The horizontal line marks the tolerance $\varepsilon = 5.10^{-3}$.

FIGURE 5.2 – Boxplots of estimation errors $\delta_{\underline{S}_u}$ for the four inputs x_3, x_4, x_5, x_6 obtained with both Variant *A.a* and Variant *A.b*. The horizontal line marks the tolerance $\varepsilon = 5.10^{-3}$.



As expected, we observe that the use of “Correlation 2” in Variant *A.b* results in lower estimation errors for these four inputs. The discrepancy observed does not stand out in Table 5.2 due to its low magnitude ($10^{-4} \sim 10^{-5}$).

Table 5.3 shows averaged estimation errors $\delta_{\underline{S}_u}$ and total number of evaluations for Variant *B*.

The results show that the use of Variant *B* leads to slightly higher estimation errors and higher failure rates than those obtained with Variant *A.a* or Variant *A.b*. However, the total number of evaluations is twice as small. As such, this approach remains interesting when one wants to estimate only first-order indices.

TABLE 5.3 – Averaged estimation errors $\delta_{\underline{S}_u}$ and total number of evaluations for Variant *B*.

input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r_{\delta_{\underline{S}_u}}$
x_1	0.5868	0.0011	0.07
x_2	0.2608	0.0013	0.08
x_3	0.0367	0.0018	0.09
x_4	0.0058	0.0013	0.1
x_5	5.10^{-5}	0.0031	0.04
x_6	5.10^{-5}	0.0036	0.05
Total number of evaluations : 32 768			

5.9.1.2 Bratley et al. function

In this second example, we consider the Bratley *et al.* function defined by,

$$f(x_1, \dots, x_d) = \sum_{i=1}^d (-1)^i \prod_{j=1}^i x_j .$$

The importance of each variable x_j depends on their own rank. More explicitly, x_1 is more influent than x_2 that is respectively more influent than x_3 and so on.

As for the g-function, Tables 5.4 and 5.5 show averaged estimation errors and total number of evaluations for Variant *A.a* and Variant *A.b*.

TABLE 5.4 – Averaged estimation errors $\delta_{\underline{S}_u}$, $\delta_{\overline{S}_u}$ and total number of evaluations for Variant *A.a*.

input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r_{\delta_{\underline{S}_u}}$	$\overline{S}_u$	$\delta_{\overline{S}_u}$	$r_{\delta_{\overline{S}_u}}$
x_1	0.6529	0.0004	0	0.7396	0.0003	0.
x_2	0.1791	0.0006	0	0.2659	0.0004	0
x_3	0.0370	0.0058	0.71	0.0764	0.0012	0.02
x_4	0.0133	0.0022	0.12	0.0343	0.0020	0.07
x_5	0.0015	0.0024	0.11	0.0062	0.0014	0
x_6	0.0015	0.0021	0.06	0.0062	0.0010	0.03
Total number of evaluations : 70 129						

Variant *A.b* gives lower estimation errors and fewer failure rates than Variant *A.a* on the four small first-indices $\underline{S}_3, \underline{S}_4, \underline{S}_5, \underline{S}_6$ which highlights the performance of “Correlation 2”. The discrepancy is particularly notable for input x_3 where Variant *A.a* reaches a failure rate of 71% against only 4% for Variant *A.b*. Furthermore,

TABLE 5.5 – Averaged estimation errors $\delta_{\underline{S}_u}$, $\delta_{\overline{S}_u}$ and total number of evaluations for Variant *A.b*.

input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r_{\delta_{\underline{S}_u}}$	$\overline{S}_u$	$\delta_{\overline{S}_u}$	$r_{\delta_{\overline{S}_u}}$
x_1	0.6529	0.0004	0	0.7396	0.0003	0
x_2	0.1791	0.0006	0	0.2659	0.0005	0
x_3	0.0370	0.0019	0.04	0.0764	0.0013	0.01
x_4	0.0133	0.0016	0.01	0.0343	0.0017	0.02
x_5	0.0015	0.0003	0	0.0062	0.0015	0.04
x_6	0.0015	0.0004	0	0.0062	0.0011	0
Total number of evaluations : 68 045						

Variant *A.b* requires less evaluations than Variant *A.a*. The boxplots represented in Figure 5.3 emphasize the latter observations.

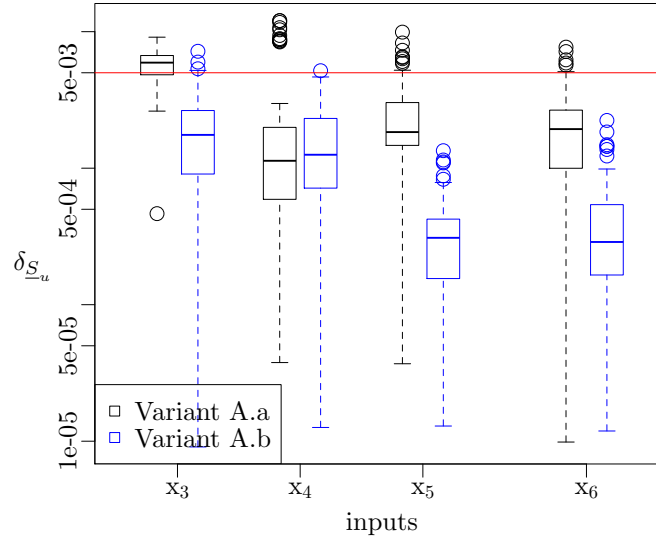
FIGURE 5.3 – Boxplots of estimation errors $\delta_{\underline{S}_u}$ for the four inputs x_3, x_4, x_5, x_6 obtained with both Variant *A.a* and Variant *A.b*. The horizontal line marks the tolerance $\varepsilon = 5.10^{-3}$.

Table 5.6 shows averaged estimation errors $\delta_{\underline{S}_u}$ and total number of evaluations for Variant *B*.

The estimation errors are lower than those of Variant *A.a* but higher than those of Variant *A.b*. Failure rates are similar to those of Variant *A.b*. Since the total number of evaluations is close to the one of Variant *A.b*, the conclusion is that Variant *B* does not bring much improvement for this example.

TABLE 5.6 – Averaged estimation errors $\delta_{\underline{S}_u}$ and total number of evaluations for Variant B .

input	$\underline{S}_u$	$\delta_{\underline{S}_u}$	$r\delta_{\underline{S}_u}$
x_1	0.6529	0.0003	0
x_2	0.1791	0.0006	0
x_3	0.0370	0.0010	0
x_4	0.0133	0.0038	0.25
x_5	0.0015	0.0024	0
x_6	0.0015	0.0026	0
Total number of evaluations : 65 536			

From the results on these two test functions, the main conclusion is that Variant $A.b$ performs the best. This highlights the efficiency of “Correlation 2” to estimate small first-order indices with quasi-Monte Carlo methods. Even if “Correlation 2” originally requires more model evaluations, this drawback is completely overridden when this estimator is included into a sequential procedure such as ours. For the case where we are only interested in estimating first-order indices, Variant B should be the best choice as illustrated with the g-function.

5.9.2 Real case model

As a real case example, we will study the payoff function of an arithmetic mean Asian call option. For a discretized Brownian motion $\mathbf{B} = (B_{t_1}, B_{t_2}, \dots, B_{t_d})$ at times $t_i = T \times i/d$, the payoff of the option is :

$$f_{\text{payoff}}(\mathbf{B}) = e^{-\rho T} \max \left(\frac{1}{d} \sum_{i=1}^d S_0 e^{(\rho - \sigma^2/2)t_i + \sigma B_{t_i}} - K, 0 \right)$$

where T is the maturity of the option, S_0 is the initial stock price, σ the volatility of the stock, ρ the interest rate, and K the strike price.

The discretized Brownian motion $\mathbf{B}$ follows a multivariate normal distribution with mean $\mathbf{0}$ and covariance matrix Σ , with $\Sigma_{ij} = \min(t_i, t_j)$. If $\mathbf{Y} \sim N(\mathbf{0}, I_d)$, given a matrix M such that $\Sigma = MM^t$, then $M\mathbf{Y} \sim N(\mathbf{0}, \Sigma)$. Hence, we can generate the discretized Brownian motion using the inverse normal cumulative distribution function $\phi^{-1} : \mathbf{B} \sim M\phi^{-1}(\mathbf{X})$ with $\phi^{-1}(\mathbf{X}) = (\phi^{-1}(X_1), \dots, \phi^{-1}(X_d))^t$ and $\mathbf{X} \sim \mathcal{U}([0, 1]^d)$. Thus, we can estimate the Sobol’ indices over $f_{\text{payoff}}(M\phi^{-1}(\mathbf{X}))$. In this particular application, the choice of M is pivotal to reduce the effective dimensionality of the model. In this regard, Sobol’ indices can help us assess whether we reduce the effective dimensionality. We will compare the case with either the Cholesky decomposition or the PCA construction.

We propose to analyze the results with averaged Sobol' estimates and averaged total number of evaluations over 100 repetitions. The tolerance ε and parameters ℓ^* , r and $\mathfrak{C}(m)$ are set as previously.

Tables 5.7, 5.8, and 5.9 show the results when M corresponds to the Cholesky decomposition of Σ .

TABLE 5.7 – Averaged values of $\hat{\underline{S}}_u$, $\hat{\bar{S}}_u$, and total number of evaluations of Variant A.a with Cholesky decomposition.

input	$\hat{\underline{S}}_u$	$\hat{\bar{S}}_u$
x_1	0.2120	0.5650
x_2	0.1370	0.4420
x_3	0.0820	0.3150
x_4	0.0408	0.1950
x_5	0.0142	0.0944
x_6	0.0048	0.0257
Total number of evaluations : 648 821		

TABLE 5.8 – Averaged values of $\hat{\underline{S}}_u$, $\hat{\bar{S}}_u$ and total number of evaluations of Variant A.b with Cholesky decomposition.

input	$\hat{\underline{S}}_u$	$\hat{\bar{S}}_u$
x_1	0.2120	0.5650
x_2	0.1370	0.4420
x_3	0.0822	0.3150
x_4	0.0434	0.1950
x_5	0.0169	0.0944
x_6	0.0049	0.0259
Total number of evaluations : 699 487		

It turns out that Variant A.b does not perform better than Variant A.a. Averaged Sobol' estimates are similar but the average total number of model evaluations is slightly worse. The explanation is that total effect Sobol' indices require more evaluations than their corresponding first-order indices. As such, "Correlation 2" falls into case 1) explained in Section 5.8.2. Variant B shows a drastic improvement on the total number of evaluations with the drawback of a false estimation for input x_4 . Its discrepancy is higher than ε when comparing to the other two variants.

TABLE 5.9 – Averaged values of $\hat{\underline{S}}_u$ and total number of evaluations of Variant B with Cholesky decomposition.

input	$\hat{\underline{S}}_u$
x_1	0.2120
x_2	0.1350
x_3	0.0774
x_4	0.0338
x_5	0.0149
x_6	0.0028
Total number of evaluations : 262 144	

Tables 5.10, 5.11, and 5.12 show the results when M corresponds to the PCA decomposition of Σ . Variant $A.a$ performs as well as Variant $A.b$ in terms of average total number of evaluations. However, Variant $A.a$ is most probably giving worse estimates for the small first-order indices. Table 5.10 shows that averaged $\hat{\underline{S}}_4$ is greater than $\hat{\bar{S}}_4$, $\hat{\underline{S}}_5$ greater than $\hat{\bar{S}}_5$, and $\hat{\underline{S}}_6$ greater than $\hat{\bar{S}}_6$, which is inconsistent with the property $\underline{S}_u \leq \bar{S}_u$. Once again, variant B shows an improvement on the total number of evaluations with the drawback of slightly overestimating the small first-order indices.

Overall, each variant captures well how the PCA Brownian motion construction reduces the high effective dimensionality of the payoff function, in contrast to the initial Cholesky decomposition.

TABLE 5.10 – Averaged values of $\hat{\underline{S}}_u$, $\hat{\bar{S}}_u$ and total number of evaluations of Variant $A.a$ with PCA decomposition.

input	$\hat{\underline{S}}_u$	$\hat{\bar{S}}_u$
x_1	0.9800	0.9960
x_2	0.0037	0.0172
x_3	0.0016	0.0023
x_4	0.0014	0.0005
x_5	0.0005	0.0001
x_6	0.0002	0.0000
Total number of evaluations : 396 221		

TABLE 5.11 – Averaged values of $\hat{\underline{S}}_u$, $\hat{\bar{S}}_u$ and total number of evaluations of Variant *A.b* with PCA decomposition.

input	$\hat{\underline{S}}_u$	$\hat{\bar{S}}_u$
x_1	0.9800	0.9960
x_2	0.0034	0.0172
x_3	0.0004	0.0023
x_4	0.0001	0.0005
x_5	0.0000	0.0001
x_6	0.0000	0.0000
Total number of evaluations : 398 648		

TABLE 5.12 – Averaged values of $\hat{\underline{S}}_u$ and total number of evaluations of Variant *B* with PCA decomposition.

input	$\hat{\underline{S}}_u$
x_1	0.9800
x_2	0.0037
x_3	0.0028
x_4	0.0029
x_5	0.0017
x_6	0.0029
Total number of evaluations : 262 144	

5.10 Conclusion

When estimating Sobol' indices, the question of how many evaluations must be performed to reach a desired precision is often raised by practitioners. This question is difficult to address mostly because the number of evaluations needed depends on the complexity of the model studied. As such, it is hard to bring out a general rule of thumb.

The sequential estimation procedure proposed in this article offers a practical solution with the construction of an estimator and error bound for Sobol' indices. The number of points is progressively augmented until the error bound becomes lower than a user specified tolerance. The procedure presented combines Sobol' sequences with either Saltelli's strategy to estimate both first-order and total effect indices, or the replication procedure to estimate only first-order indices. Furthermore, we investigated the use of a recent estimator well-suited to the estimation

of small first-order indices using quasi-Monte Carlo methods. The efficiency of this estimator, called “Correlation 2”, was assessed and highlighted on two test functions. Overall, the variant combining Saltelli’s strategy and “Correlation 2” gave the best results, with low failure rates across all indices.

As a future project, the same estimators and algorithms can be designed for rank-1 lattices using the results in [RH16], and noticing that rank-1 lattices of the same size are also replicated designs of order 1.

acknowledgements

This work is supported by :

- the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020),
- the United States National Science Foundation (grant DMS-1522687).

The authors thank Fred J. Hickernell and Clémentine Prieur for initiating this collaborative work and Elise Arnaud for her proofreading. The authors are grateful to Stephen Joe, Frances Y. Kuo and Art B. Owen for their helpful answers and suggestions.

Sensitivity analysis and optimisation of a Land Use and Transport Integrated model

Les résultats de ce chapitre ont donné lieu à un article soumis pour publication [GACP16]. L'objectif de ce chapitre est d'apporter une réponse à la problématique P_2 soulevée en introduction portant sur l'estimation des indices de Sobol' dans le cas de variables de sortie vectorielles du modèle. Ce chapitre fait également l'objet d'une application sur le modèle transport-urbanisme Tranus.

Ce chapitre propose une procédure de calage du modèle transport-urbanisme Tranus [dlB99]. Le calage d'un modèle transport-urbanisme joue un rôle central permettant à la fois d'estimer les valeurs optimales des paramètres (variables d'entrée) et d'obtenir un modèle robuste et fiable pour tester des scénarios prédictifs. Les approches traditionnelles de calage misent en oeuvre sur de tels modèles s'appuient en majorité sur des procédures économétriques *ad hoc*.

La procédure de calage que nous proposons dans cet article comprend deux étapes principales : une phase d'analyse de sensibilité et une phase d'optimisation.

La méthode d'analyse de sensibilité proposée est une extension de la méthode répliquée au cas de sorties vectorielles. Cette extension s'appuie sur la généralisation des indices de Sobol' proposée dans [GJKL14]. Ces derniers indices sont construits à partir d'une décomposition de Hoeffding multidimensionnelle de la sortie vectorielle du modèle.

La phase d'optimisation est une procédure itérative hybride combinant deux approches : stochastique et déterministe. L'approche stochastique correspond à l'algorithme EGO (Efficient Global Optimisation) introduit par Mockus *et al.* [MTZ78]. L'approche déterministe est une formulation analytique de l'optimum pour un type particulier de paramètre. Les résultats obtenus avec notre méthode de calage sont comparés à des premiers résultats obtenus par une méthode de calage plus classique mêlant techniques d'économétrie et approches essai-erreurs pouvant durer plusieurs semaines.

Dans ce chapitre, nous proposons également une procédure bootstrap pour la construction d'intervalles de confiance pour l'estimation des indices de Sobol' avec la méthode répliquée dans le cas de sorties vectorielles. Cette contribution, non décrite dans [GACP16], est détaillée dans la section 6.B.

Titre : Sensitivity analysis and optimisation of a Land Use and Transport Integrated model.

Auteurs : Laurent Gilquin^{a,b}, Elise Arnaud^b, Thomas Capelle^a, Clémentine Prieur^b.

a Inria Grenoble - Rhône-Alpes, Inovallée, 655 avenue de l'Europe, 38330 Montbonnot

*b Univ. Grenoble Alpes, Jean Kuntzmann Laboratory, F-38000 Grenoble, France
CNRS, LJK, F-38000 Grenoble, France, Inria*

6.1 Introduction

Land use and transport integrated (LUTI) models have received a regain of interest from researchers and urban planners the last decade. Among the large number of available LUTI models, we focus here on the model *Tranus* developed by [dIB99]. *Tranus* and other LUTI models aim to represent the strong interactions between travel behaviours and land use. Their scope of use ranges from urban metropolitan areas to regional level. LUTI modelling is mainly used to evaluate alternative planning scenarios, simulating their impact on land cover and travel demand. Instantiating a LUTI model requires the gathering of huge amounts of data and the estimation of several parameters to reproduce as closely as possible base year observations (such as socio-economic surveys, transport data, etc ...) on the studied area. These models include systems of complex nonlinear equations. Analysing these systems is typically a hard task, particularly in the presence of parameters whose effects may be difficult to assess. Interactions between parameters of the model make that a small change in a parameter may result in large changes in the model outputs. In such cases, calibration plays a central role, as it helps us to estimate optimal values of these parameters, creating a robust model. The standard calibration approach of these models relies on econometric ad-hoc procedures and trial and errors techniques. An exception can be found in [AH00], where an automatic calibration of the LUTI model *MEPLAN* is proposed.

Assessing sensitivity of the input parameters on the outputs during the calibration process is essential to reach a proper calibration of the model and ensure better predicting capabilities. As a matter of fact, in the context of traffic simulation [DBH14] put forward the need to carry out a sensitivity analysis as a first step preceding the calibration of the model. Global sensitivity analysis methods are useful tools to quantify the influence of the model inputs on the outputs and detect potential interactions between them. Among the large number of available approaches, the variance based method introduced by [Sob93] allows to calculate sensitivity indices called Sobol' indices. These indices are scalars between 0 and 1 that summarise the influence of each input (or set of inputs) : the higher is the index the more influential

is the input (or set of inputs). First-order indices estimate the main effect from each input whereas higher-order indices estimate the corresponding order of interactions between inputs. Various estimation procedures of these indices have been proposed in the literature (see [SCS08] for a survey). A first implementation of a sensitivity analysis on Tranus was performed in [DSA⁺12] using the “pick-freeze” estimation procedure introduced in [Sob93]. Unfortunately, this procedure requires a significant number of model evaluations that increases depending on the input space dimension. A solution to break this dependency lies in the use of replicated designs. Based on such designs, an estimation procedure for Sobol’ indices was proposed in [MJ08] and further studied in [TP15]. We propose here an extension of the replication procedure to deal with multidimensional output. This generalisation is then applied to the land use and activity module of Tranus to select the most influential parameters based on main effects and second-order interactions.

Following the sensitivity analysis, we propose an estimation procedure of the influential parameters of the land use and activity module using a two-stage iterative optimisation. Firstly, a stochastic algorithm is applied to find optimal values of the influential parameters selected by the sensitivity analysis. Secondly, an analytical optimisation of an internal dispersion parameter is performed taking as inputs the previously optimised parameters. This two-stage optimisation is iterated until an equilibrium is reached on the internal dispersion parameter. This second step is based on a careful investigation of the system of equations detailed in [CSV15].

The rest of the paper is organised as follows. In Section 6.2, a detailed description of Tranus is provided. In particular, we focus on the land use and activity module detailing the principal variables relevant to this paper. Section 6.3 details the two main ingredients of our calibration procedure from a methodological point of view : sensitivity analysis and the stochastic algorithm used in the two-stage iterative optimisation. In Section 6.4, we detail our whole calibration procedure in the form of an algorithm. Our methodology is then applied to the study area of North Carolina. Our results are compared to those obtained with a standard ad-hoc procedure.

6.2 Description of Tranus and problem Statement

6.2.1 General structure of the model

In this paper, we will consider Tranus, a land use and transportation integrated model (LUTI). This type of software provides a framework for modelling land use and transportation in an integrated manner. It provides a flexible package to be used from urban, regional up to national scale. Tranus takes the basis from the classic input-output model (see [LS63]) and generalises it, adding the transportation layer on top of it.

The area of study is divided in spatial zones and economical sectors. The concept of sectors is more general than in the traditional definition. It may include the standard sectors in which the economy is divided (agriculture, manufacturing, mining,

etc.), factors of production (capital, land and labour), population groups, employment, floorspace, land, energy, or any other that is relevant to the spatial system being represented.

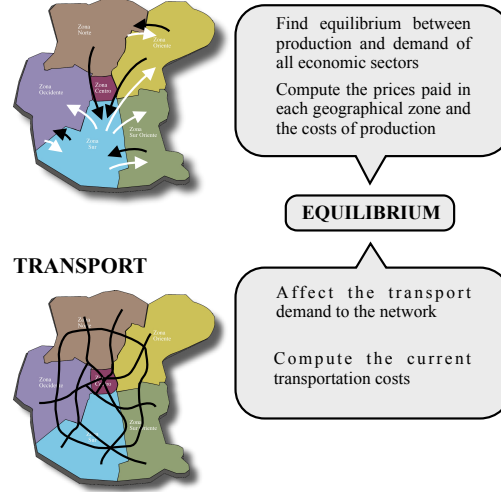
Tranus combines a land use and activity module with a transportation module. Both of these modules are linked together and serve as input to each other, as illustrated in Figure 6.1. The spatial economic system is simulated by the activity model, representing the interactions of the various economical sectors in a specific time period. These interactions result in transportation demand that is afterwards affected by the transportation module. In this way the movements of people or freight are explained as the results of the economic and spatial interaction between activities, the transport system and the real estate market. In return, the transport demand and the flux of goods influences the activities in the territory, affecting the access to transportation, the price of goods and ultimately the land rents. Both of these modules are based on the classic discrete choice theory (see [McF73, MT00]), input-output analysis (see [Leo41]), land choice (see [Wil81]), multi-modal path choice and trip assignment. A comprehensive review of transport modelling can be found in [OW11].

The convergence is attained when both modules are in equilibrium. The land use and activity module iteratively equilibrates supply and demand, also computing the consumption costs and prices. This is done at current transportation costs. In the other hand, the transportation module assigns the transport demand to the network and computes the new transportation costs. This back and forth procedure iterates until a general equilibrium condition is found. This condition is basically that neither land use, nor transport evolves anymore. Figure 6.1 illustrates this mechanic. As it is already well calibrated, no further details on the transportation module of Tranus are provided. The procedure proposed in this paper focuses exclusively on the land use and activity module.

6.2.2 The land use and activity module

The land use and activity module's objective is to find an equilibrium between the production and demand of all economic sectors and zones of the modeled region. To attain the equilibrium, various parameters and functions are used to represent the behaviour of the different economic agents. Among these parameters are demand elasticities, attractiveness of geographical zones, technical coefficients, etc.

It is important for our paper to make the difference between two types of economic sectors; transportable and non-transportable sectors. The main difference between these two types of sectors, is that transportable sectors can be consumed in a different place from where they were produced. As an example, the demand for coal from a metal industry can be satisfied by a mining industry located in another region. On the other hand, a typical non-transportable sector is floorspace : land is consumed where it is "produced". Transportable sectors *generate flow*, that induces transport demand, which ultimately influence transportation costs. Non-

FIGURE 6.1 – Schematic overview of Tranus
LAND-USE

transportable sectors, on the other hand, neither require transportation nor generate flows. In [CSV15] we proposed a detached calibration of Tranus, calibrating first non-transportable sectors, and building up the rest of the calibration over them. We will consider this version of Tranus for the rest of the paper.

In the following, we introduce terminologies used in Tranus that are relevant to this paper. The set of equations relevant to this paper can be found in the Appendix. For a complete description of Tranus, see [dlB99]. We will consider a modelled territory or region divided in a set $\mathcal{Z}$ of spatial zones, and the economy to be represented by a set $\mathcal{N}$ of economical sectors. We will consider also that we have data $Y_0 = \{Y_{0i}^n\}_{n \in \mathcal{N}, i \in \mathcal{Z}}$ corresponding to the observed production for a base year in each zone and for each economical sector. We denote as sub-index 0 this year of reference for the calibration.

The basis of the land use and activity model relies in four principal quantities :

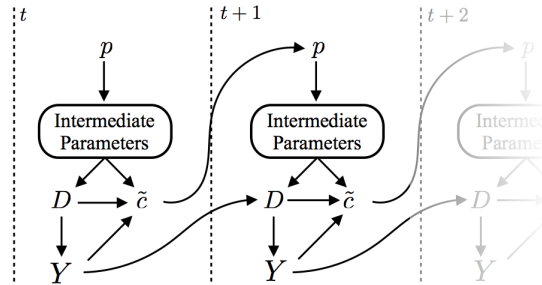
- Productions : $Y = \{Y_i^n\}_{n \in \mathcal{N}, i \in \mathcal{Z}}$ expresses how many “items” of an economic sector n are produced in a zone i .
- Demands : $D = \{D_i^{mn}\}_{(m,n) \in \mathcal{N} \times \mathcal{N}, i \in \mathcal{Z}}$ expresses how many items of a sector n are demanded by the part of sector m located in zone i .
- Prices : $p = \{p_i^n\}_{n \in \mathcal{N}, i \in \mathcal{Z}}$ defines the price of (one item of) sector n located in zone i .
- Costs : $\tilde{c} = \{\tilde{c}_i^n\}_{n \in \mathcal{N}, i \in \mathcal{Z}}$ is the cost of consumption of sector n in zone i

Productions, demands, prices and costs are defined for each sector and each zone. Each possesses an economic representation that varies depending on the type of sector considered. To fix ideas, productions of a household sector represent the

total households of a certain income level (rich for instance) in the corresponding zone. When considering as sector a certain type of housing, production can also represent the available surface (square meters) in the zone of interest. Demands are tied to productions and represent how many units of the desired economic sector are consumed in a zone by another economic sector. The canonical example is demand for housing by households. Prices are very straightforward, representing the price of consuming one unit of the given economical sector in that zone. When considering e.g., households, prices represent salaries/incomes. Finally, costs are results of the model after the consumption chain has taken place, the equilibrium state is attained when costs are in perfect equilibrium with prices, and demand are in equilibrium with productions.

The computation of each of these quantities depends on one another and are linked by group of non-linear equations. The multiple dependencies of these variables are represented in Figure 6.2. The feedback is as follows, at current prices, after the computation of various intermediate variables (that will not be detailed here) we compute the demand and costs of consumption, finally computing the production. This production is not necessarily equal to the demand, and the costs of production are not necessarily equal to the prices, so we have to iterate these quantities until an equilibrium in production-demand and costs-prices is found. To do so, the costs of iteration t are fed as prices in iteration $t + 1$, and production in iteration t is compared to demand in iteration $t + 1$. All above computations are repeated until convergence is obtained in these two sets of variables. This process is done from bottom (land, exogenous prices) to top (transportable sectors).

FIGURE 6.2 – Sketch of computations in the land use and activity module. t is the iteration index of the convergence.



6.2.3 Problem statement

The calibration procedure that we propose consists in adjusting the model parameters to be able to reproduce a base year's data in the study area. It is important to note that our calibration procedure is applied independently for each sector n generating flow. We will consider this type of sectors because it is where a sensitivity analysis is relevant, as the many interactions with the transportation costs and different parameters make calibration very complex. Also, for this particular

sectors, modellers would like to identify the relevant parameters to improve their calibration. In [CSV15], a decomposed calibration of Tranus is proposed, calibrating first the non-transportable sectors and later the transportable sectors. Starting from the results of [CSV15], we decided that the calibration of the transportable sectors and the numerous parameters that are often ignored as their interactions are not fully understood, could benefit from a sensitivity analysis. The calibration is as following, once a sector generating flow is selected, the model parameters relative to this sector are adjusted through our calibration procedure. The process is repeated for each sector generating flow. The justification of this approach relies on a sector-wise decomposition of the system of equations of the land use and activity module. Further details concerning this decomposition are provided in [CSV15].

Given initial values of the parameters for a sector $n \in \mathcal{N}$ generating flow, the land use and activity module estimates the adjustment parameters $h^n = (h_i^n)_{i \in \mathcal{Z}}$ of the utilities (see Appendix A, (6.13)), known as shadow prices. In economics, the utility is defined as a measure of preference over sets of goods and services. For example, in the land use module, utility is used to measure the willingness of a household to relocate from a geographic zone to an other. Still in economics, the shadow price is formally defined as the value of the Lagrange multiplier at the optimal solution of a constrained optimisation problem. However, in our problem the definition of the shadow price differs as it only represents an adjustment factor added to the price to ease the convergence of the land use module. Thus here, the shadow prices are price correcting additive factors. They compensate the utilities to replicate the base year production Y_0 . The following optimisation problem is solved to compute the shadow prices :

$$\hat{h}^n = \arg \min_{h^n} \|Y(h^n) - Y_0\|^2, \quad (6.1)$$

where $\|\cdot\|^2$ denotes the $\mathbb{L}^2$ norm. For the rest of the paper, this optimisation of the shadow price is viewed as an internal process of the land use and activity model. Figure 6.3 gives a scheme of the inputs and outputs considered for each sector n generating flow. The inputs parameters fall into different categories :

- parameter λ^n is involved in Equation (6.13) of Annexe 6.A,
- parameters β^n is involved in Equation (6.14) of Annexe 6.A,
- parameters b_l^n , $l \neq n, l \in \mathcal{N}$, are involved in Equation (6.15) of Annexe 6.A.

Parameter λ^n represents the marginal utility of income, in common terms, it is the weight of prices in the utility function. The β^n coefficient, is the logit dispersion parameter of the location probabilities. Finally, the b_l^n coefficient, is the weight of sector l in the attractor for sector n , the later are very important to transportable sectors

The outputs considered are built upon a new quantity called normalised shadow price. The normalised shadow price $\tilde{h}_i^n$ corresponds to the percentage of the price

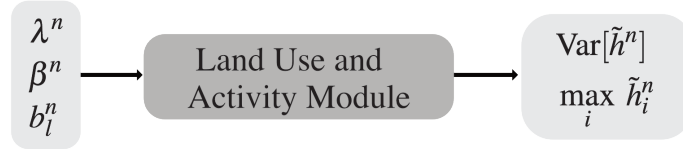
p_i^n corrected by the shadow price h_i^n , that is :

$$\tilde{h}_i^n = 100 \times \left| \frac{h_{n,i}}{p_{n,i}} \right|, \quad i \in \mathcal{Z} \quad (6.2)$$

where $|\cdot|$ denotes the absolute value function. We set $\tilde{h}^n = (\tilde{h}_i^n)_{i \in \mathcal{Z}}$, the vector of normalised shadow prices relative to the sector n . The two outputs considered are the following :

- i) the variance of the normalised shadow prices : $\text{Var}[\tilde{h}^n]$
- ii) the maximum of the normalised shadow prices : $\max_{i \in \mathcal{Z}} \tilde{h}_i^n$

FIGURE 6.3 – Inputs and outputs of the land use and activity module for the sector n generating flow.



For each sector n generating flow, a good calibration would be one that results in small values of the normalised shadow prices particularly in terms of variance. Indeed, minimising the variance of the normalised shadow prices is a general consensus reached by both modellers and users of Tranus.

In the next section, we present the two main methodological ingredients used in our calibration procedure : replication procedure for the sensitivity analysis and EGO algorithm for the stochastic optimisation. Each tool is presented in a general framework nonspecific to Tranus. The algorithm summarising our calibration procedure is presented in Section 6.4.1.

6.3 Calibration procedure's main tools

6.3.1 Global Sensitivity Analysis : replication procedure for multivariate outputs

In this section, we present a generalisation of the replication procedure, proposed in [MJ08] and deeply studied in [TP15], to the case of multidimensional output. We first review the generalisation of the variance-based method introduced by [Sob93] that relies on the estimation of generalised Sobol' indices (see [LMM11, GJKL14]).

Consider the following model :

$$f: \begin{cases} \mathbb{R}^d & \rightarrow \mathbb{R}^m \\ x = (x_1, \dots, x_d) & \mapsto z = f(x) \end{cases}$$

where f is the model, z the output vector, x the input vector, d the dimension of the input space and m the dimension of the output space.

Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. The uncertainty on the inputs is modeled by a random vector $X = (X_1, \dots, X_d)$ whose components are independent. Denote by Z the vector of random variables modelling the output vector :

$$Z = (Z_1, \dots, Z_m) = f(X_1, \dots, X_d).$$

Let $P_X = P_{X_1} \otimes \dots \otimes P_{X_d}$ denote the probability distribution of X . Suppose that $f \in \mathbb{L}^2(P_X)$ and that the covariance matrix of Z , denoted by Σ , is positive definite. Let u be a subset of $\{1, \dots, d\}$ and denote by $\sim u$ its complementary. We set $X_u = (X_i)_{i \in u}$ and $X_{\sim u} = (X_i)_{i \in \{1, \dots, d\} \setminus u}$. Recall the following [Hoe48] decomposition of f :

$$f(X) = f_0 + f_u(X_u) + f_{\sim u}(X_{\sim u}) + f_{u, \sim u}(X_u, X_{\sim u}), \quad (6.3)$$

where $f_0 = \mathbb{E}[Z]$, $f_u = \mathbb{E}[Z|X_u] - f_0$, $f_{\sim u} = \mathbb{E}[Z|X_{\sim u}] - f_0$ and $f_{u, \sim u} = Y - f_u - f_{\sim u} - f_0$. By taking the covariance matrix of each side of (6.3), due to orthogonality we get :

$$\Sigma = C_u + C_{\sim u} + C_{u, \sim u} \quad (6.4)$$

Let Id_m be the $m \times m$ identity matrix. Equation (6.4) can be projected on a scalar as follows :

$$\text{Tr}(\text{Id}_m \Sigma) = \text{Tr}(\text{Id}_m C_u) + \text{Tr}(\text{Id}_m C_{\sim u}) + \text{Tr}(\text{Id}_m C_{u, \sim u}) \quad (6.5)$$

where Tr denotes the trace operator. Following (6.5) and under the condition $\text{Tr}(\Sigma) \neq 0$, the generalised Sobol' index is defined as follows :

$$S^u(f) = \frac{\text{Tr}(C_u)}{\text{Tr}(\Sigma)}. \quad (6.6)$$

The generalised Sobol' index $S^u(f)$ is a scalar between 0 and 1 that summarises the influence of inputs in u on the output Z . The closer to 1 is the index $S^u(f)$, the more influential is the set u . An index equal to 0 means that the set u is not correlated to the output Z . First-order indices estimate the main effect from each input whereas higher-order indices estimate the corresponding order of interactions between inputs.

Remark 6.1. When $u = (v, w)$ is a 2-subset of $\{1, \dots, d\}$, the influence of the interaction between v and w is quantified by the second-order generalised Sobol' index defined by : $S^{(v, w)}(f) - S^v(f) - S^w(f)$.

Standard estimation of $S^u(f)$ The standard estimation procedure for $S^u(f)$ is a generalisation of the one used in the univariate case (see [Sob93]). The procedure consists of a Monte-Carlo pick-freeze method. In the pick-freeze method, the Sobol index is viewed as the regression coefficient between the output of the model and

its pick-freezed replication. This replication is obtained by holding the value of the variable of interest X_u (frozen variable) and by resampling the other variables $X_{\sim u}$ (picked variables).

We set $Z = f(X_u, X_{\sim u})$ and $Z^u = f(X_u, X'_{\sim u})$ where $X'_{\sim u}$ is an independent copy of $X_{\sim u}$. Let $N > 0$ be an integer and $Z_1, \dots, Z_N$ (resp. $Z_1^u, \dots, Z_N^u$) be N independent copies of Z (resp. Z^u) where :

$$Z_i = (Z_{i,1}, \dots, Z_{i,m}), \quad Z_i^u = (Z_{i,1}^u, \dots, Z_{i,m}^u), \quad \forall i \in \{1, \dots, N\}.$$

As in [MNM06] and [JKL⁺14], the following estimator of $S^u(f)$ is constructed :

$$\widehat{S}^u(f) = \frac{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N Z_{i,l} Z_{i,l}^u - \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l} + Z_{i,l}^u}{2} \right)^2 \right)}{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l}^2 + (Z_{i,l}^u)^2}{2} - \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l} + Z_{i,l}^u}{2} \right)^2 \right)}. \quad (6.7)$$

Using this approach, estimating all first-order Sobol' indices requires $N(d+1)$ evaluations of the model through $d+1$ designs of experiments each of size N . In the univariate case, the replication procedure introduced in [MJ08] allows to estimate all first-order indices with only two replicated designs each of size N resulting in a total of $2 \times N$ evaluations of the model. Replicated designs are also referred as *plans based on permuted columns* in [McK95], [MMM06] and [MMM08]. In these papers, an arbitrary number of r replications of the initial design is used to define an estimator of first-order indices. This estimator is of different nature from the one introduced in [MJ08] and further studied (asymptotic properties for first-order indices) and generalised in [TP15] to the estimation of closed second-order indices. We propose here an extension of this procedure to the case of multivariate output. With this new approach, the number of model evaluations required to compute all first-order or second-order generalised Sobol' indices can be drastically reduced.

Replication procedure for $S^u(f)$ The replication method relies on the construction of two replicated designs of experiments $\mathbf{X}$ and $\mathbf{X}'$ defined as follows :

$$\mathbf{X} = \begin{pmatrix} X_{1,1} & \dots & X_{1,j} & \dots & X_{1,d} \\ \vdots & & \vdots & & \vdots \\ X_{i,1} & \dots & X_{i,j} & \dots & X_{i,d} \\ \vdots & & \vdots & & \vdots \\ X_{N,1} & \dots & X_{N,j} & \dots & X_{N,d} \end{pmatrix}, \quad \mathbf{X}' = \begin{pmatrix} X'_{1,1} & \dots & X'_{1,j} & \dots & X'_{1,d} \\ \vdots & & \vdots & & \vdots \\ X'_{i,1} & \dots & X'_{i,j} & \dots & X'_{i,d} \\ \vdots & & \vdots & & \vdots \\ X'_{N,1} & \dots & X'_{N,j} & \dots & X'_{N,d} \end{pmatrix},$$

where $\forall k \in \{1, \dots, d\}$, $X_{1,k}, \dots, X_{N,k}$ are N independent copies of X_k . For the estimation of first-order indices, $\mathbf{X}$ and $\mathbf{X}'$ are two replicated Latin Hypercubes. For the estimation of closed second-order indices, $\mathbf{X}$ and $\mathbf{X}'$ are two replicated randomised orthogonal arrays (see [TP15] for further details on the construction of

$\mathbf{X}$ and $\mathbf{X}'$). Denote by $\mathbf{Z}$ and $\mathbf{Z}'$ the two arrays of model outputs associated to these two designs. Let Z_i and Z'_i denote their corresponding rows, $\forall i \in \{1, \dots, N\}$ we have :

$$\begin{aligned} Z_i &= f(X_{i,1}, \dots, X_{i,d}) = (Z_{i,1}, \dots, Z_{i,m}), \\ Z'_i &= f(X'_{i,1}, \dots, X'_{i,d}) = (Z'_{i,1}, \dots, Z'_{i,m}). \end{aligned}$$

The key point of the replication method consists in a “smart” rearrangement of the rows of $\mathbf{Z}'$ to mimic the pick-freeze method. The array resulting from this rearrangement corresponds to $\mathbf{Z}^u$. In the pick-freeze method, for each u , the evaluation of $\mathbf{Z}^u$ requires a new design of experiments. At the opposite, in the replication method $\mathbf{Z}^u$ requires no additional evaluations of the model. Let π denote the permutation used to re-arrange $\mathbf{Z}'$, $\forall i \in \{1, \dots, N\}$:

$$Z_i^u = f(X'_{\pi(i),1}, \dots, X'_{\pi(i),d}) = (Z'_{\pi(i),1}, \dots, Z'_{\pi(i),m}).$$

Let $u = \{u_1, \dots, u_s\} \subset \{1, \dots, d\}$. From a design point of view, π is chosen to insure that :

$$X'_{\pi(i),u_j} = X_{i,u_j}, \quad \forall j \in \{1, \dots, s\}, \forall i \in \{1, \dots, N\}.$$

thus insuring that both $\mathbf{Z}$ and $\mathbf{Z}^u$ are evaluated on the same u coordinates. We propose here to estimate $S^u(f)$ using formula (6.7) with both $\mathbf{Z}$ and $\mathbf{Z}^u$. In this fashion, we extend the replication procedure to the estimation of generalised first- and closed second-order Sobol' indices. For a concern of size of the paper, we choose to not further detail the replication procedure. The interested reader can find a detailed description in [TP15].

The selection of an influential parameter is made by fixing a threshold for both first- and second-order indices. If one of the estimated indices is higher than the threshold, the corresponding parameter is selected. For the application presented in Section 6.4, we subjectively fixed the threshold equals to 0.1.

6.3.2 Stochastic optimisation : EGO algorithm

The stochastic optimisation procedure presented in this section corresponds to the Efficient Global Optimisation (EGO) algorithm introduced by [MTZ78]. The main idea underlying the EGO algorithm is to fit a response surface, often denoted by metamodel, to data collected by evaluating the complex numerical model at a few points. The metamodel is then used in place of the numerical model to optimise the parameters. The metamodel used in the EGO algorithm is a Gaussian process defined as follows :

$$g: \begin{cases} \mathbb{R}^d & \rightarrow \mathbb{R} \\ x = (x_1, \dots, x_d) & \mapsto z = g(x) = \mu(x) + \varepsilon(x) \end{cases}$$

where x are the parameters selected with the sensitivity analysis, z a scalar output of the numerical model, d the dimension of the input space, μ the model trend and ε

is a centered stationary Gaussian process $\varepsilon(x) \sim N(0, K_\chi)$. χ denotes the structure of the covariance matrix K_χ of ε . Let x^i, x^j denote two points of $\mathbb{R}^d$, $\chi = \{r, \theta, \sigma\}$ with $(K_\chi)_{i,j} = \sigma^2 r_\theta(x^i - x^j)$ where :

- $r_\theta(\cdot)$ is the correlation function chosen here to be the Matérn 5/2 function,
- σ^2 is the variance of g ,
- θ are the hyperparameters of r .

The parameters μ , σ and θ are estimated by maximum likelihood. In the following, Z denotes the random variable modelling the output z .

Expected Improvement Once the metamodel is fitted, it is used by the algorithm to search for a minimum candidate. The EGO algorithm uses a searching criterion called “expected improvement” that balances local and global search. Let x be a candidate point, the expected improvement evaluated at x writes as follows :

$$EI_\chi(x) = E[\max(z_{min} - Z, 0)],$$

where z_{min} is the current minimum of the metamodel. A numerical expression of $EI_\chi(x)$ can be derived. Let $\hat{Z}$ denote the *BLUP* (Best Linear Unbiased Estimator) estimator (see [JSW98]) of Z and $\sigma_{\hat{Z}}$ its standard deviation, the following expression for $EI_\chi(x)$ is obtained :

$$EI_\chi(x) = (z_{min} - \hat{z}(x))\phi_{\mathcal{N}}\left(\frac{z_{min} - \hat{z}(x)}{\sigma_{\hat{Z}}}\right) + \sigma_{\hat{Z}}f_{\mathcal{N}}\left(\frac{z_{min} - \hat{z}(x)}{\sigma_{\hat{Z}}}\right)$$

where $\phi_{\mathcal{N}}$ is the normal cumulative distribution function and $f_{\mathcal{N}}$ is the normal probability density function. The first term of $EI_\chi(x)$ is a local minimum search term whereas the second term corresponds to a global search of uncertainty regions.

The main steps of the EGO algorithm can be summarised as follows :

1. generate a design of experiments and evaluate the numerical model on these points,
2. fit the metamodel with both the design of experiments and the associated model outputs,
3. search a new evaluation point using the expected improvement criterion,
4. evaluate the numerical model on this new point and re-estimate the parameters of the meta-model (θ, σ) ,
5. repeat steps 3 to 5 until a stopping criterion is reached.

For the choice of the stopping criterion, one can look at the value of the expected improvement. Indeed, a value of the expected improvement close to zero indicates that the input space has been sufficiently explored. Thus, an upper bound on the expected improvement can be selected as the stopping criterion. Here, we set the lower bound equal to 10^{-5} . Thus, the stopping criterion writes :

$$EI_\chi(x) \leq 10^{-5}$$

To ensure that the EGO algorithm finishes, we also fix a maximum number of iterations equal to 200. The two R packages “DiceOptim” and “DiceDesign” developed by [RGD12] are used to implement the EGO algorithm.

6.4 Application to Tranus

In this section, we detail our calibration procedure for the land use and activity module of Tranus in the form of an algorithm. The procedure includes both methods presented in Section 6.3. Our methodology is then applied to the Mississippi region including the Chickasaw, Lee, Pontotoc, and Union Counties, including the area's four largest towns, which are Houston, Tupelo, Pontotoc, and New Albany.

6.4.1 Calibration procedure algorithm

Algorithm 6.1 summarises our calibration procedure for the land use and activity module of Tranus. Recall that the calibration procedure is applied independently for each sector n generating flow.

Algorithm 6.1 Calibration procedure for the land use and activity module

- 1: **for** each sector n generating flow **do**
 - 2: Set : $\lambda^{n(0)} \leftarrow \lambda_0^n$
 - 3: Run sensitivity analysis with inputs : $\beta^n, \{b_l^n\}_{n \neq l}$ and outputs : $Var[\tilde{h}^n], \max_{i \in \mathcal{Z}} \tilde{h}_i^n$
 - 4: Instantiate :
 - $\rho^{n(0)} \leftarrow$ set of most influential parameters,
 - $k \leftarrow 1$
 - 5: **while** $|\lambda^{n(k)} - \lambda^{n(k-1)}| \geq \varepsilon$ **do**
 - 6: Given $\lambda^{n(k-1)}$, estimate $\rho^{n(k)}$ with the EGO algorithm
 - 7: Given $\rho^{n(k)}$, estimate $\lambda^{n(k)}$ with the analytical optimisation
 - 8: **return** optimal values ρ^{n*} and λ^{n*}
-

Once a sector n generating flow is selected, the sensitivity analysis presented in Section 6.3.1 is performed on the parameters β^n and $b_{n,l}$, $l \neq n$. The outputs considered for the sensitivity analysis are both the variance and the maximum of the normalised shadow prices $\tilde{h}^n$ (Equation 6.2). The set of influential parameters selected is denoted by ρ^n .

Following the sensitivity analysis, an iterative optimisation is conducted. This optimisation comprises two stages. At iteration k , the EGO algorithm presented in Section 6.3.2 is applied to find optimal values of the set $\rho^{n(k)}$ given $\lambda^{n(k-1)}$. The parameters not selected by the sensitivity analysis are fixed at their initial value. Then, an analytical optimisation of $\lambda^{n(k)}$ is performed taking as inputs the optimal values found for the set $\rho^{n(k)}$. This step is further described in the next section. The process is iterated until an equilibrium is reached on λ^n . At the end of the iterations optimal values ρ^{n*} and λ^{n*} are returned.

Remarks

1. One might wonder why the sensitivity analysis does not include the parameter λ^n . The main reason behind this choice is that the analytical optimisation

provides a global optimum for λ^n . Hence by automatically selecting the parameter λ^n , we both reduce the number of Sobol' indices to estimate and obtain a finer optimisation of the land use and activity module.

2. For the stochastic optimisation, we chose to conserve only one output to perform the EGO algorithm : the variance of the normalised shadow prices : $\text{Var}[\tilde{h}^n]$. As already mentioned, minimising the variance of the normalised shadow prices over the maximum is a general consensus reached by both the modeller and users of Tranus.

6.4.2 Analytical optimisation procedure

Consider a sector n generating flow. In this section, we present an analytical procedure that estimates the parameter λ^n . For the sake of clarity, the notation $\lambda^n, \lambda^{n(k)}$ and $\beta^n, \beta^{n(k)}$ are confounded for the rest of the section.

The parameter λ^n is involved in the location probabilities equation (Equation 6.13 of Appendix 6.A) :

$$U_{ij}^n = \lambda^n(p_j^n + h_j^n) + t_{ij}^n, \quad (i, j) \in \mathcal{Z}^2. \quad (6.8)$$

The optimal value of λ^n cannot be retrieved directly from Equation 6.8 as the quantity $(p_j^n + h_j^n)$ is estimated as a whole during the internal optimisation of the shadow prices (see [CSV15] for further details). To overcome this problem, we introduce an auxiliary variable :

$$\phi_j^n = \lambda^n(p_j^n + h_j^n), \quad j \in \mathcal{Z}.$$

With this new variable, Equation 6.8 can be rewritten as follows :

$$U_{ij}^n = \phi_j^n + t_{ij}^n, \quad (i, j) \in \mathcal{Z}^2. \quad (6.9)$$

Recall that the shadow prices are prices correcting additive factors that are calibrated to obtain a small variance. From Equation 6.9, we can express the optimal value of λ^n that minimises the variance of the shadow prices. We set $\phi^n = (\phi_j^n)_{j \in \mathcal{Z}}$ with all other parameters fixed, in particular the parameters ρ^n estimated with the EGO algorithm (see Section 6.3.2). The corresponding calibration problem writes :

$$\phi^{n*} = \arg \min_{\phi^n} \|Y(\phi^n) - Y_0\|^2. \quad (6.10)$$

Recall that we have $p^n = (p_j^n)_{j \in \mathcal{Z}}$ and $h^n = (h_j^n)_{j \in \mathcal{Z}}$ the vectors of prices and shadow prices. Once the optimal value ϕ^{n*} is obtained, the equilibrium prices p^{n*} can be computed solving a linear system. Then, the shadow prices are expressed as follows :

$$h^n = \frac{\phi^{n*}}{\lambda^n} - p^{n*}.$$

From this, the following problem can be posed :

$$\min_{\lambda^n} \text{Var} \left[\frac{\phi^{n*}}{\lambda^n} - p^{n*} \right]$$

where the analytical solution equals $\lambda^{n*} = \frac{\text{Var}(\phi^{n*})}{\text{Cov}(\phi^{n*}, p^{n*})}$. Note that its value can be negative. The details of Problem (6.10) and how the prices are computed can be found in [CSV15].

6.4.3 Results

In this section, we propose an application of our calibration procedure to a study area in Mississippi region. We first give a brief insight of the area of study. Then, we present the results of both the sensitivity analysis and the iterative optimisation used in our calibration procedure. The results are compared to those obtained with a former ad-hoc procedure.

Study of area : The area of study is the Mississippi region divided into 12 economical sectors and 103 spatial zones. The dataset was provided by Brian Morton¹. In Table 6.1 are listed the different economical sectors of Tranus relative to the area of study. Our calibration procedure focus only on the sectors generating flow.

TABLE 6.1 – Economical sectors of Tranus for the study area of Mississippi.

sector n	name	description	type
1	AFFHM(Agriculture,Fishing,...)	Business	Exogenous
2	Commercial	Business	Transportable
3	Other industries	Business	non-Transportable
4	Single person (15-64 years old)	Households	Transportable
5	Married couple with kids	Households	Transportable
6	Married couple without kids	Households	Transportable
7	Other familie with kids	Households	Transportable
8	65 years and older	Households	Transportable
9	All other households	Households	Transportable
10	1-unit housing units	Floorspace	Land
11	Multiunit housing units	Floorspace	Land
12	Mobile homes	Floorspace	Land

Results of the sensitivity analysis : A total of 7 sensitivity analyses are performed, one for each sector generating flow. For each sensitivity analysis, the 12 following parameters are considered :

1. Brian Morton is a senior research associate working at the Center for Urban and Regional Studies of North Carolina, Hickerson House, 108 Battle Lane, Campus Box 3410. The data were provided to us during an informal workshop taking place at INRIA on the topic of LUTI model calibration.

- The parameter β^n .
- The 11 parameters $b_l^n, l \neq n$.

In Table 6.2 are listed the distribution of each parameter. These distributions were selected by expertise. The Sobol' index S^k will refer to the parameter labeled by k . The outputs considered are the variance and the maximum of the normalised

TABLE 6.2 – Distributions of the 12 parameters, $U(a, b)$ stands for the uniform distribution with support $[a, b]$

parameters	labels	distributions
β^n	1	$U(2, 10)$
$\{b_l^n\}_{l=1, \dots, n, l \neq n}$	$2, \dots, 12$	$U(0, 1)$

shadow prices introduced in Section 6.2.3. The approach proposed is to use the replication procedure presented in Section 6.3.1 to estimate first-order and second-order generalised Sobol' indices of these parameters. Asymptotic confidence intervals can be computed for first-order Sobol' indices (see [TP15]). Using Remark 6.1, bootstrap confidence intervals can be derived for second-order Sobol' indices. The formulation of the bootstrap confidence intervals is detailed in Annexe 6.B.

Before presenting the main results, we propose to illustrate the selection procedure of the influential parameters for a sector. Figures 6.4 shows the results obtained for the estimation of first-order and second-order indices relative to Sector 4. The dashed line represents the threshold value used for selecting the influential parameters.

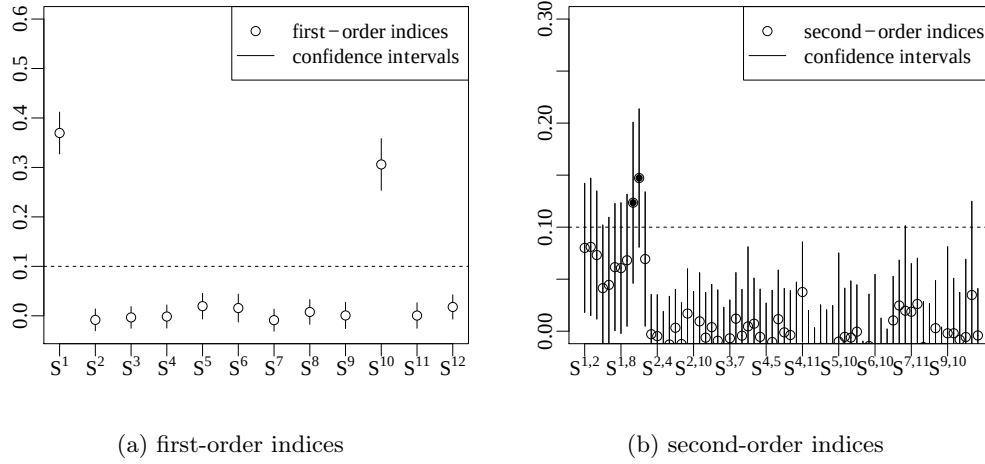
For the estimation of first-order indices, a size $N = 5 \times 10^3$ was chosen for the two replicated Latin Hypercubes required by the replication procedure (Section 6.3.1). Looking at the results, the parameters β^4 and b_{10}^4 are the most influential. Since the sum of the first-order is less than 75% is it interesting to study the second-order indices.

For the estimation of second-order indices. A size $N = 67^2$ was selected for the two replicated randomised orthogonal arrays required by the replication method. The two black points of graph (b) of Figure 6.4 correspond to the two most influential interactions : $\beta^4 * b_{10}^4$ and $\beta^4 * b_{11}^4$. The number of bootstrap replications used to compute the confidence intervals equals 1000.

As a conclusion, only 3 of the 12 parameters of the sector 4 are significantly influential either directly by their main effects or by their second-order interactions : β^4, b_{10}^4 and b_{11}^4 .

The same procedure is performed for the other sectors generating flow. For each sector n , the set ρ^n comprised of the most influential parameters selected by the sensitivity analysis is listed in Table 6.3. The last column of the table gives the proportion of the variance explained by the selected parameters. This proportion is

FIGURE 6.4 – First- and second-order indices estimation for the 12 parameters of sector 4.



calculated by multiplying the sum of the generalised Sobol' indices of the first two columns by 100. Looking at the results, only 3 parameters appear to be overall the most influential : β^n , b_{10}^n and b_{11}^n , $n \in \{2, 4, 5, 6, 7, 8, 9\}$.

TABLE 6.3 – Most influential parameters selected by the sensitivity analysis based on main effects and second-order interactions

sector n	first-order	second-order	selected parameters : ρ^n	variance explained (in percentage)
2	β^2	none	β^2	33
4	β^4, b_{10}^4	$\beta^4 * b_{10}^4, \beta^4 * b_{11}^4$	$\beta^4, b_{10}^4, b_{11}^4$	95
5	β^5, b_{10}^5	$\beta^5 * b_{10}^5, \beta^5 * b_{11}^5$	$\beta^5, b_{10}^5, b_{11}^5$	89
6	$\beta^6, b_{10}^6, b_{11}^6$	$\beta^6 * b_{10}^6$	$\beta^6, b_{10}^6, b_{11}^6$	90
7	β^7, b_{10}^7	none	β^7, b_{10}^7	85
8	β^8, b_{10}^8	$\beta^8 * b_{10}^8$	β^8, b_{10}^8	89
9	$\beta^9, b_{10}^9,$	$\beta^9 * b_{10}^9$	β^9, b_{10}^9	93

These results fall within our range of expectation. The parameter β^n is a dispersion parameter of a multinomial logit function (see Equation 6.14 of Annexe 6.A). A slight variation of this parameter leads to a significant change in the calculation of the probabilities of localisation. Both parameters b_{10}^n and b_{11}^n act as weights in the attractiveness for sector n . These two parameters are more prone to be influential than the other b_l^n since sectors 10 and 11 correspond to the two main floorspace.

Looking at the last column of Table 6.3, for some sectors (in particular for sector 2) the variance is not fully explained by the parameters selected. A solution to further explained this variance would consist in using Saltelli's procedure to estimate first-order, second-order and total-effect Sobol' indices (index measuring the contribution of an input through all its interactions). However, Saltelli's procedure would requires $14 \times N$ evaluations of the model for each sector generating flow. In comparison, our approach require only $4 \times N$ evaluations of the model to estimate both first-order and closed second-order indices ($2 \times N$ for first-order indices and $2 \times q^2$, $q \approx \sqrt{N}$ for closed second-order indices). Since the number of evaluations was a concern for us, we decided to go for the replication procedure at the price of not using the total effect Sobol' indices.

A compromise would be to fit a metamodel and evaluate first-order and total effect Sobol' indices with the metamodel outputs. Such a methodology has been applied by [GCM14] to the traffic simulation model Aimsun. In their application, the use of a metamodel is particularly efficient as microscopic traffic simulators are known to be quite time-consuming. However, we argue that the complexity of Tranus makes it hard to perform a metamodel based estimation of Sobol' indices.

An alternative to the calculation of Sobol' indices was proposed by [GM14] to perform a sensitivity analysis prior to the calibration of the traffic model VISSIM. This approach called quasi-OTEE is a screening method based on the original Elementary Effects method introduced by [Mor91]. The cost of the quasi-OTEE method writes $(m - n + 1) * (m + n) / 2$ where m is the number of Morris trajectories and n is the number of quasi-optimal trajectories selected among the m latter. The authors advise to choose m and n in the respective ranges $[500, 1000]$ and $[10, 20]$. With the minimal setting $m = 500$ and $n = 10$, the cost of the quasi-OTEE method equals 125 205 which is 6 times higher than the cost of our application to estimate first- and second-order indices ($2 \times 5000 + 2 \times 67^2$). Furthermore, as a screening method the quasi-OTEE approach should be less precise than the calculation of Sobol' indices.

Results of the iterative optimisation For each sector n generating flow, the aim is to find the set of parameters (ρ^n, λ^n) minimising the variance of $\tilde{h}^n$. The initial value λ_0^n (Step 3 of Algorithm 6.1) instantiating the parameter λ^n is obtained by expertise. The results obtained in terms of variance and maximum of the normalised shadow prices are compared to those obtained with a former ad-hoc procedure. The number of initial evaluations performed to fit the metamodel for the set of parameters (ρ^n, λ^n) of each sector n is the following :

- 21 evaluations for sector 2,
- 51 evaluations for sector 4 to 6.
- 41 evaluations for sector 7 to 9.

The quality of the fitting is assessed by diagnostic plots (fitted values against response values, standardized residuals, qqplot of standardized residuals) based on leave-one-out cross validation results (see [RGD12] for further details). For each sector n , the evaluations include the one for the optimal set of parameters obtained

with the ad-hoc procedure.

Table 6.4 summarises the results obtained with both the ad-hoc procedure and our iterative optimisation. ρ^{n*} and λ^{n*} denote the optimal values of the parameters obtained at the end of both approaches for each sector n generating flow. The column gain represents the improvement (in percentage) of the variance obtained with our iterative estimation relatively to the one obtained with the ad-hoc procedure conducted by experts.

TABLE 6.4 – Variance and maximum of the normalised shadow prices $\tilde{h}_n$ obtained with both ad-hoc procedures and our iterative optimisation

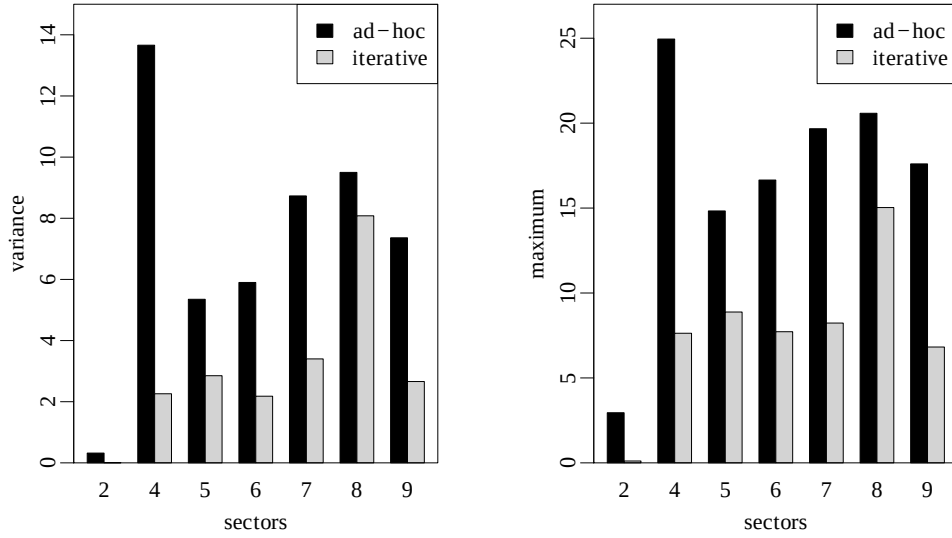
sector n	procedure	ρ^{n*}	λ^{n*}	variance $\tilde{h}_n$	max $\tilde{h}_n$	gain
2	ad-hoc	2	0.005	0.32	2.95	98 %
	iterative	4.03	0.43	7×10^{-3}	0.11	
4	ad-hoc	(2, 0, 0)	0.001	13.66	24.95	83 %
	iterative	(6.49, 0.38, 0)	0.001	2.26	7.63	
5	ad-hoc	(2, 0, 0)	0.001	5.35	14.83	47 %
	iterative	(2.50, 0.02, 0.79)	-0.013	2.85	8.88	
6	ad-hoc	(2, 0, 0)	0.001	5.90	16.65	63 %
	iterative	(6.64, 0.05, 0.79)	-0.003	2.18	7.72	
7	ad-hoc	(2, 0)	0.001	8.73	19.67	61 %
	iterative	(9.17, 1)	0.001	3.40	8.23	
8	ad-hoc	(2, 0)	0.001	9.50	20.58	15 %
	iterative	(5.72, 0.97)	0.001	8.08	15.03	
9	ad-hoc	(2, 0)	0.001	7.36	17.6	64 %
	iterative	(9.29, 0.95)	0.001	2.66	6.82	

Looking at the results, we observe that the values of the variance and maximum of the normalised shadow prices obtained with the ad-hoc procedure are heterogeneous. Furthermore, the value of the maximum is quite high for some sectors (up to 20% of the price). The results obtained with our iterative optimisation are relatively homogeneous except for sectors 2 and 8. The discrepancy observed for these two sectors comes from the quality of their respective datasets. Indeed, the data relative to commercial business (sector 2) are easy to collect thus of high quality and quantity. At the opposite, data relative to the 65 years and older households (sector 8) are quite complex to collect and often lacking of precision.

The main observation is that an improvement in terms of both variance and maximum of the normalised shadow prices is observed for all sectors while using our approach. Figure 6.5 gives an illustration of this improvement. The black bars represent the values obtained with the ad-hoc procedure, the grey bars those obtai-

ned with our iterative approach. A significant diminution for both the variance and maximum criteria is observed.

FIGURE 6.5 – Variance and maximum of the normalised shadow prices obtained with both the ad-hoc procedure (referred as ad-hoc) and our iterative optimisation (referred as iterative)



(a) variance of the normalised shadow prices (b) maximum of the normalised shadow prices

Furthermore and most importantly our approach is drastically faster than the ad-hoc procedure conducted by experts. Our calibration procedure requires a few hours compared to several days (up to weeks) for the ad-hoc procedure.

As a final remark, we decided here to conserve only the best set of parameters (ρ^{n*}, λ^{n*}) obtained with our calibration procedure. [CA14] proposed an alternative where a metamodel is fitted and several best sets of parameters are selected. It is true that for complex systems such as LUTI models, the best solution of the EGO optimisation probably corresponds to only one of many combinations of the inputs that provide the model with a sufficient robustness. The method of [CA14] allows to investigate the behavior of the model for various combinations and to derive uncertainty margins of the outputs. Adapting this methodology to our calibration procedure of Tranus would be an interesting complementary work.

Conclusion

In this paper we proposed a calibration procedure combining a global sensitivity analysis and an iterative optimisation to calibrate the land use and activity module

of the LUTI model Tranus. We presented as sensitivity analysis a generalisation of the replication procedure to select the most influential parameters of the model when the output is multidimensional. The optimisation phase was carried on using a two stages process combining stochastic and deterministic procedures. An application to the study area of Mississippi was presented. Our methodology was compared to a former ad-hoc calibration procedure in terms of variance and maximum of the normalised adjustment parameters (shadow prices). Our approach showed a significant improvement on both criteria reducing the value of the variance by a large margin with a drastic gain of time. These results have proved that our methodology is useful to outperform calibration of such models. The next step would consist in verifying if the optimal values found for the parameters ensure better predicting capabilities when evaluating alternative planning scenarios.

Acknowledgments

The authors are thankful to Brian J. Morton for providing high quality data and instantiating the Tranus model on the study area of Mississippi. This work is supported by the CITiES project funded by the Agence Nationale de la Recherche (grant ANR-12-MONU-0020).

6.A Tranus system of equations

Only a subset of model equations relevant to this paper is presented. The demand is computed for all combinations of zone i , demanding (consuming) sector m and demanded sector n :

$$D_i^{mn} = (Y_i^{*m} + Y_i^m) a_i^{mn} S_i^{mn} \quad (6.11)$$

$$D_i^n = D_i^{*n} + \sum_m D_i^{mn} \quad (6.12)$$

where Y_i^{*m} is the given exogenous production (for exports), Y_i^m the induced endogenous production obtained in the previous iteration (or initial values), and D_i^{*n} exogenous demand. D_i^n in (6.12) then gives the total demand for sector n in zone i . a_i^{mn} is a technical demand coefficient and S_i^{mn} is the substitution proportion of sector n when consumed by sector m on zone i .

In parallel to demand, one computes the **utility** of all pairs of production and consumption zones, i and j :

$$U_{ij}^n = \lambda^n (p_j^n + h_j^n) + t_{ij}^n . \quad (6.13)$$

Here, λ^n is the marginal utility of income for sector n and t_{ij}^n represents transport disutility. Since utilities and disutilities are difficult to model mathematically (they include subjective factors such as the value of time spent in transportation), Tranus incorporates adjustment parameters h_j^n , so-called shadow prices, amongst the model parameters to be estimated.

From utility, we compute the probability that the production of sector n demanded in zone i , is located in zone j . Every combination of n , i and j is computed :

$$Pr_{ij}^n = \frac{A_j^n \exp(-\beta^n U_{ij}^n)}{\sum_k A_k^n \exp(-\beta^n U_{ik}^n)} . \quad (6.14)$$

Here, k ranges over all zones. β^n is the dispersion parameter for the multinomial logit model expressed by the above equation. A_j^n represents attractiveness of zone j for sector n and is expressed as follows :

$$A_j^n = \sum_l b_n^l Z_j^l \quad (6.15)$$

where the $b_{n,l}$ are the relative weights of sector l in the attractive function of sector n and the Z_j^l are variables depending on prices and productions.

From these probabilities, new productions are then computed for every combination of sector n , production zone j and consumption zone i :

$$Y_{ij}^n = D_i^n Pr_{ij}^n . \quad (6.16)$$

Total production of sector n in zone j , is then :

$$Y_j^n = \sum_i Y_{ij}^n \quad (6.17)$$

$$= \sum_i D_i^n Pr_{ij}^n . \quad (6.18)$$

The set of prices are also governed by a set on non-linear equations, and are computed simultaneously to attain equilibrium.

6.B Bootstrap confidence intervals

We first review the definition of bootstrapping as introduced in [ET93]. Consider an estimator $\hat{T}$ of a quantity of interest T function of a random variable W . This estimator writes :

$$\hat{T}(W_1, \dots, W_N),$$

where $W_1, \dots, W_N$ are N independent random variables distributed as W . The basic idea of bootstrapping is to select an integer $R > 0$ and for $r = 1, \dots, R$, randomly sample with replacement an N -set $(W_1^r, \dots, W_N^r)$ from $(W_1, \dots, W_N)$. Then, the r bootstrap replicate of $\hat{T}$ is obtained by :

$$\hat{T}^r = \hat{T}(W_1^r, \dots, W_N^r). \quad (6.19)$$

Denote by $\mathcal{R} = \{\hat{T}^1, \dots, \hat{T}^R\}$ the set of all bootstrap replicates of $\hat{T}$. Confidence intervals for T can be build based on $\mathcal{R}$. These confidence intervals are called bootstrap confidence intervals. We detail below two main form of bootstrap confidence intervals : classic and gaussian.

Classic. The confidence interval is obtained by taking its lower and upper bounds to be the quantiles of order $\alpha/2$ and $1 - \alpha/2$ of $\mathcal{R}$:

$$UC = \left[\hat{T}^{\alpha/2}, \hat{T}^{1-\alpha/2} \right].$$

Gaussian. The confidence interval is obtained by approximating the quantiles of order $\alpha/2$ and $1 - \alpha/2$ of $\mathcal{R}$ respectively by $m - q_\alpha \sigma$ and $m + q_\alpha \sigma$ where :

- m and σ are the mean and standard deviation of $\mathcal{R}$,
- q_α is the quantile of order $1 - \alpha/2$ of a standard normal distribution.

The resulting confidence interval writes :

$$UC = [m - q_\alpha \sigma, m + q_\alpha \sigma].$$

In our application, we aim to construct classic bootstrap confidence intervals for the generalised second-order Sobol' indices defined in Remark 6.1 by the difference :

$$S^{(v,w)}(f) - S^v(f) - S^w(f), \quad (v, w) \in \mathcal{D}^2.$$

Recall that for $j \in \mathcal{D}$, the estimator $\widehat{S}^j(f)$ is defined by :

$$\widehat{S}^j(f) = \frac{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N Z_{i,l} Z_{i,l}^j - \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l} + Z_{i,l}^j}{2} \right)^2 \right)}{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l}^2 + (Z_{i,l}^j)^2}{2} - \left(\frac{1}{N} \sum_{i=1}^N \frac{Z_{i,l} + Z_{i,l}^j}{2} \right)^2 \right)},$$

where $Z_i = (Z_{i,1}, \dots, Z_{i,m})$ and $Z_i^j = (Z_{i,1}^j, \dots, Z_{i,m}^j)$. For the first-order indices $S^j(f)$, $j \in \mathcal{D}$, classic bootstrap confidence intervals are obtained with equation (6.19) where $\hat{T} = \widehat{S}^j(f)$ and $W_i = (Z_i, Z_i^j)$ for $i = 1, \dots, N$.

For $(j, k) \in \mathcal{D}^2$, $j < k$, the estimator $\widehat{S}^{(j,k)}(f)$ is defined by :

$$\widehat{S}^{(j,k)}(f) = \frac{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N \mathcal{Z}_{i,l} \mathcal{Z}_{i,l}^{(j,k)} - \left(\frac{1}{N} \sum_{i=1}^N \frac{\mathcal{Z}_{i,l} + \mathcal{Z}_{i,l}^{(j,k)}}{2} \right)^2 \right)}{\sum_{l=1}^m \left(\frac{1}{N} \sum_{i=1}^N \frac{\mathcal{Z}_{i,l}^2 + (\mathcal{Z}_{i,l}^{(j,k)})^2}{2} - \left(\frac{1}{N} \sum_{i=1}^N \frac{\mathcal{Z}_{i,l} + \mathcal{Z}_{i,l}^{(j,k)}}{2} \right)^2 \right)},$$

where $\mathcal{Z}_i = (\mathcal{Z}_{i,1}, \dots, \mathcal{Z}_{i,m})$ and $\mathcal{Z}_i^{(j,k)} = (\mathcal{Z}_{i,1}^{(j,k)}, \dots, \mathcal{Z}_{i,m}^{(j,k)})$. Classic bootstrap confidence intervals for $S^{(j,k)}(f)$ are obtained with equation (6.19) where $\hat{T} = \widehat{S}^{(j,k)}(f)$ and $W_i = (\mathcal{Z}_i, \mathcal{Z}_i^{(j,k)})$ for $i = 1, \dots, N$.

Hence, for the second-order indices, classic bootstrap confidence intervals are obtained with equation (6.19) where $\hat{T} = \widehat{S}^{(j,k)}(f) - \widehat{S}^j(f) - \widehat{S}^k(f)$ and $W_i = (Z_i, \mathcal{Z}_i, Z_i^j, Z_i^k, \mathcal{Z}_i^{(j,k)})$ for $i = 1, \dots, N$. The notations Z and $\mathcal{Z}$ highlight that in our application the designs used for the estimation of first-order indices differ from those used for the estimation of closed second-order indices.

Conclusion et perspectives

Conclusion

Dans cette thèse nous nous sommes principalement intéressé à des extensions de la méthode répliquée. La méthode répliquée permet d'estimer tous les indices de Sobol' d'ordre un ou tous les indices de Sobol' fermés d'ordre deux à l'aide de seulement deux plans d'expériences. Pour une précision donnée, cette approche permet de réduire considérablement le nombre d'évaluations du modèle nécessaires à l'estimation de ces indices.

Les extensions que nous avons proposées aux chapitres 3, 4 et 6 ont pour but de faire face à des problématiques communes rencontrées lors d'applications sur des modèles numériques : présence de corrélations entre les variables d'entrée du modèle, sorties vectorielles, estimation récursive des indices. Le développement de ces extensions repose sur des modifications des plans d'expériences utilisés au sein de la méthode répliquée : hypercubes latins et tableaux orthogonaux de force deux.

Nous avons également proposé au chapitre 5 une borne d'erreur pour l'estimation des indices de Sobol' par la méthode de Saltelli et par la méthode répliquée lorsqu'on utilise un échantillonnage de type quasi-Monte Carlo. Cette borne d'erreur sert d'indicateur pour stopper l'algorithme itératif proposé dans ce chapitre.

Le chapitre 6 contextualise certains de nos résultats théoriques à l'étude du modèle transport-urbanisme Tranus. La méthodologie que nous proposons présente un apport conséquent pour l'étude de l'influence des paramètres et pour le calage des paramètres les plus influents de Tranus, par rapport aux approches plus basiques appliquées jusqu'à présent sur ce type de modèles. D'autre part, ces dernières approches nécessitent généralement plusieurs semaines avant l'obtention de résultats contre quelques heures pour notre méthodologie.

Perspectives

Nous présentons trois perspectives à ce travail de thèse. Ces perspectives portent toutes sur la méthode répliquée.

Dans [Tis12], les auteurs proposent une étude asymptotique de la méthode répliquée pour l'estimation des indices d'ordre un. Une extension intéressante serait d'étendre cette étude au cas des indices d'ordre deux. Au chapitre 6, nous proposons la construction d'intervalles de confiance bootstrap pour l'estimation des indices

d'ordre un et deux par la méthode répliquée. Une conséquence directe de l'étude asymptotique serait la construction d'intervalles de confiances asymptotiques.

Comme mentionné dans [Tis12], le point de départ pour cette étude est l'article [Loh08] contenant des résultats intéressants sur les propriétés asymptotiques des tableaux orthogonaux de force deux. Ensuite, l'idée consiste à généraliser les outils développés de [TP15] au cas de l'estimation des indices d'ordre deux.

La seconde perspective consiste à établir un critère d'optimalité pour la construction de plans destinés à l'estimation des indices de Sobol' par la méthode répliquée. Les plans d'expériences utilisés actuellement (hypercubes latins, séquences digitales pour l'ordre un, tableaux orthogonaux randomisés pour l'ordre deux) possèdent des propriétés intéressantes en terme de variance pour l'estimation de la moyenne μ d'un modèle f . Développer un critère directement pour l'estimation des indices de Sobol' est un enjeu qui n'a pas encore, à notre connaissance, de solution satisfaisante actuellement.

Dans les travaux que nous avons présentés, tous les indices d'ordre un sont estimés à partir de deux hypercubes latins répliqués, puis tous les indices d'ordre deux fermés sont estimés à partir de deux tableaux orthogonaux randomisés de force deux répliqués. Il semble naturel de se poser la question de l'estimation de tous les indices d'ordre un et deux à partir de deux plans répliqués seulement. En effet, on constate que deux tableaux orthogonaux randomisés de force deux répliqués définis comme à la formule (2.14) page 51, notés $\mathcal{P}$ et $\mathcal{P}'$, sont à la fois des plans répliqués d'ordre deux mais également des plans répliqués d'ordre un. Ainsi, $\mathcal{P}$ et $\mathcal{P}'$ permettraient théoriquement d'estimer à l'aide de la méthode répliquée à la fois tous les indices d'ordre un et tous les indices fermés d'ordre deux. Cependant, si les couples d'entrées (X_j, X_k) sont discrétisés en q^2 valeurs distinctes, les marginales X_j , elles, ne sont discrétisées qu'en q valeurs distinctes, chacune répétée q fois. Il existe donc plusieurs permutations π_j garantissant que $\mathcal{P}$ et $\mathcal{P}'$ soit deux plans répliqués d'ordre un au sens de la définition 2.5 page 51. Une perspective consiste donc à proposer un critère pour choisir pour chaque $j \in \mathcal{D}$ une permutation π_j permettant d'éviter des effets de confusion (*aliasing*) entre l'estimation de l'indice d'ordre un S_j et les estimations des différents indices d'ordre deux $\underline{S}_{\{j,k\}}$, $\forall k \in \mathcal{D}, k > j$.

ANNEXE A

Additional notes

This appendix addresses major questions raised by Art B. Owen. These questions focus on the presence of unwanted duplicates when estimating first- or closed second-order Sobol' indices with the replication procedure (section 2.3 page 47), and on the economics of using orthogonal arrays.

A.1 Unwanted duplicates

We first study the presence of unwanted duplicates when estimating first-order indices with the replication procedure.

A.1.1 Case of first-order indices

Problem statement

Recall that the estimation of first-order indices with the replication procedure relies on the construction of two replicated Latin hypercubes (equation 2.13 page 48) denoted by $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ and $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ where $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,d})$, $\mathbf{X}'_i = (X'_{i,1}, \dots, X'_{i,d})$. The study we present below is conducted on the discrete structure of the two Latin hypercubes : $X_{i,k} = \pi_k^{(1)}(i)$ et $X'_{i,k} = \pi_k^{(2)}(i)$, $k \in \{1, \dots, d\}$.

For $j \in \{1, \dots, d\}$, the estimation of the first-order index S_j is performed by ordering the rows of $\mathcal{P}'$ with the permutation π_j such that $X_{i,j} = X'_{\pi_j(i),j}$. The problem raised is the following : for $k \neq j$ such an ordering can lead to an unwanted duplicate $X_{i,k} = X'_{\pi_j(i),k}$. We call collision such an event. Collision is also referred as “coincidence defect” in [Owe91]. If a collision occurs, rows $\mathbf{X}_i$ and $\mathbf{X}'_{\pi_j(i)}$ have more than one coordinate in common and confounding effects will arise between S_j and higher order indices.

One can model the distribution of the number of collisions for a pair of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$, $i \in \{1, \dots, n\}$. Let $k \neq j$, $k \in \{1, \dots, d\}$, $X'_{\pi_j(i),k}$ is a random variable with values in $\{1, \dots, n\}$. The probability that a collision occur in position k for the pair of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ writes :

$$\sum_{l=1}^n \mathbb{P} \left[X_{i,k} = X'_{\pi_j(i),k} \mid X_{i,k} = l \right] \mathbb{P} [X_{i,k} = l] = \sum_{l=1}^n \frac{1}{n} \times \frac{1}{n} = \frac{1}{n}.$$

Since the $d - 1$ random variables $\{X'_{\pi_j(i),k}\}_{k=1,\dots,d,k \neq j}$ are independent, the number of collisions for the pair of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ is binomial with parameters $d - 1$ and $1/n$. Let C be the random variable modeling this number of collisions :

$$P[C = l] = \binom{n}{l} \left(\frac{1}{n}\right)^l \left(1 - \frac{1}{n}\right)^{n-l}, \quad l \in \{0, \dots, d - 1\}.$$

Table A.1 below contains values of $P[C \geq 1]$ (one collision or more) for some values of d and n . This table gives an overview of the number of collisions for all n pairs

TABLE A.1 – Values of $P[C \geq 1]$ (one collision or more) for some values of d and n .

$d \setminus n$	50	100	250	500	750	1000
5	0.077	0.039	0.016	0.008	0.005	0.004
10	0.166	0.086	0.035	0.018	0.012	0.009
15	0.246	0.131	0.055	0.028	0.019	0.014
20	0.319	0.174	0.073	0.037	0.025	0.019
25	0.384	0.214	0.092	0.047	0.032	0.024
30	0.443	0.253	0.110	0.056	0.037	0.028

of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}_{i=1}^n$. As an example, if $d = 30$ and $n = 1000$, in average 28 pairs among the 1000 will lead to collisions.

One important aspect to mention is the following. Even if a large number of pairs of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ lead to collisions, the positions k where collisions occur may differ from one pair to the other. As a result, instead of a large confusion effect between S_j and a higher order index, we obtain a multitude of small confusion effects between S_j and many higher order indices. This most certainly is the origin of why in practice we do not observe any bias in the estimation of S_j .

Nonetheless, we propose in the following a solution to ensure that no collision occur during the estimation of S_j , $j \in \{1, \dots, d\}$.

Solution

We propose here to construct deux Latin hypercube without unwanted duplicates. Our solution relies on a specific choice of permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}$ and $\pi_1^{(2)}, \dots, \pi_d^{(2)}$ generating the two replicated Latin hypercubes as in (2.13). With such a construction, for $j \in \{1, \dots, d\}$, the permutation π_j ordering the rows of $\mathcal{P}'$ writes :

$$\pi_j = \pi_j^{(2)-1} \circ \pi_j^{(1)}.$$

The underpinning idea to our solution is the use of cyclic permutations, also referred as n -cycles. Let Π_n be the set of all permutations of $\{1, \dots, n\}$. A n -cycle π is an

element of Π_n satisfying :

$$\begin{cases} \pi^n = \underbrace{\pi \circ \pi \circ \dots \circ \pi}_{n \text{ times}} = I_n, \\ \pi^\alpha(i) \neq \pi^\beta(i), \quad (\alpha, \beta, i) \in \{1, \dots, n\}^3, \alpha \neq \beta, \end{cases} \quad (\text{A.1})$$

where I_n is the identity permutation and $\circ$ is the composition law in Π_n . There exist exactly $(n-1)!$ distinct n -cycles in Π_n . The construction of a n -cycle can be performed with Sattolo's algorithm [Sat86]. This algorithm generates a random cyclic permutation uniformly distributed in the set of the $(n-1)!$.

Once a n -cycle π is constructed, we apply the following mapping between powers of π and permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}, \pi_1^{(2)}, \dots, \pi_d^{(2)}$:

$$\forall j \in \{1, \dots, d\}, \begin{cases} \pi_j^{(1)} = \pi^{d-j} \\ \pi_j^{(2)} = \pi^{d+j} \end{cases} \quad (\text{A.2})$$

Note that such a mapping requires to choose $n > 2d$ to avoid that permutation $\pi_j^{(1)}$ collide with a permutation $\pi_{j'}^{(2)}$. Furthermore, for a Latin hypercube, the probability to obtain this configuration writes $\frac{(n-1)!}{(n!)^d}$ where $(n-1)!$ is the choice of the n -cycle and $(n!)^d$ is the number of Latin hypercube with d columns.

We prove below that mapping (A.2) ensures that for each permutation π_j , no collision occur between each pair of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$. First, remark that with mapping (A.2) π_j writes :

$$\pi_j = \pi_j^{(2)-1} \circ \pi_j^{(1)} = \pi^{n-(d+j)+d-j} = \pi^{-2j}.$$

Then, *ad absurdum*, suppose a collision occurred in position $k \neq j$, that is $X_{i,k} = X'_{\pi_j(i),k}$. In other words :

$$\pi_k^{(1)}(i) = \pi_k^{(2)} \circ \pi_j(i),$$

that is :

$$\pi^{d-k}(i) = \pi^{d+k-2j}(i).$$

Then, from (A.1) :

$$\begin{aligned} \pi^{d-k}(i) = \pi^{d+k-2j}(i) &\Rightarrow d-k = d+k-2j, \\ &\Rightarrow k = j. \end{aligned}$$

Or we assumed $k \neq j$ which leads to the contradiction. As a result, the mapping (A.2) guarantees that no collision will occur for the estimation of each first-order Sobol' index S_j . A futur work would be to study the effect of mapping (A.2) on the estimator of S_j . On one hand, the space from which the cyclic permutation π is sampled is smaller than Π_n . On the other hand, mapping (A.2) does not guarantee anymore that permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}$ and $\pi_1^{(2)}, \dots, \pi_d^{(2)}$ are independent. The futur work would be to characterize the bias resulting from this mapping.

A.1.2 Case of closed second-order indices

The case of closed second-order indices is treated analogously to the previous case. Let $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ be the two replicated randomized orthogonal arrays of strength two defined by (2.14) page 51. The estimation of the closed second-order index $S_{\{k,l\}}$ is performed by ordering the rows of $\mathcal{P}'$ with the permutation $\pi_{\{k,l\}}$ such that $X_{i,\{k,l\}} = X'_{\pi_{\{k,l\}}(i),\{k,l\}}$.

Through a reasoning similar to the one detailed in the previous section, one can show that for a pair of rows $\{\mathbf{X}_i, \mathbf{X}'_{\pi_{\{k,l\}}(i)}\}$, the number of collisions is binomial with parameters $d - 2$ et $1/q$. We denote once again by C this number. Table A.2 contains values of $P[C \geq 1]$ (one collision or more) for some d and q .

TABLE A.2 – Values of $P[C \geq 1]$ (one collision or more) for some d and q .

$d \setminus q$	7	11	16	23	29	32
5	0.370	0.249	0.176	0.125	0.100	0.091
10	0.709	0.533	0.403	0.299	0.245	0.224
15	0.865	0.710	0.568	0.439	0.366	0.338
20	0.938	0.820	0.687	0.551	0.468	0.435
25	0.971	0.888	0.773	0.640	0.554	0.518
30	0.987	0.931	0.836	0.712	0.626	0.589

This time, we observe that the probability of one or more collisions occurring is way higher than in the previous case. The justification of this observation is the following. Given n the number of rows of $\mathcal{P}$ et $\mathcal{P}'$, in the case of first-order indices each input is discretized into n values against only $q \approx \sqrt{n}$ values in the case of closed second-order indices.

However, as noted in the previous section, all collisions do not occur in the same positions. Even if the number of collisions is relatively high, in practice we do not observe any sharp bias in the estimation of closed second-order indices.

For closed second-order indices, finding a solution to reduce the number of collisions is not trivial. That is mainly due to columns of orthogonal arrays not being identified as permutations. Solving this problem is left as a future work.

A.2 Orthogonal arrays and gain

In this section, we propose to detail the economics of using orthogonal arrays for the estimation of closed second-order indices. Recall that the estimation of these indices with the replication procedure relies on the construction of two replicated randomized orthogonal arrays of strength two (equation 2.14).

The cost of this procedure writes $2q^2$ where q is the level parameter of each orthogonal array. The choice of q must satisfy the constraint $q \geq d - 1$. As a consequence, the cost of the replication procedure is at least $2(d - 1)^2$. The cost of the classic procedure proposed by Sobol' is $n \left(\frac{d(d - 1)}{2} + 1 \right)$. We compare below these two costs given two configurations : either d is small or large.

For small values of d , one may choose $n = q^2 > (d - 1)^2$. Let D be the difference between the cost of the replication procedure and the cost of the classic procedure :

$$D = n \left(1 - \frac{d(d - 1)}{2} \right).$$

Since the estimation of closed second-order indices is only relevant for $d \geq 3$, we have $D < 0$.

For high values of d , we may have to choose $q = d - 1$. In this case, D writes :

$$\begin{aligned} D &= 2(d - 1)^2 - n \left(\frac{d(d - 1)}{2} + 1 \right), \\ D &= (d - 1)^2 \left(2 - \frac{n}{2} \right) - \frac{n(d + 1)}{2}. \end{aligned}$$

$D < 0$ for $n > 4$ which is always guaranteed to happen (choosing $n \leq 4$ is equivalent to construct designs of at most 4 points).

The same comparison can be made between the replication procedure and the first strategy of Saltelli (theorem 2.1). The cost of Saltelli's strategy to estimate first-order and total effect indices is $n(d + 2)$. The cost of the replication procedure to estimate first- and closed second-order indices is $2n + 2q^2$, $q \geq d - 1$. The difference between these two costs equals :

$$D = 2q^2 - nd.$$

For small values of d , we may choose $n = q^2$, $q > d - 1$, and D writes :

$$D = n(2 - d).$$

The replication procedure seems more advantageous in this case.

For high values of d , we set $q = (d - 1)$, the difference becomes :

$$D = (d - 1)(2(d - 1) - n) - n.$$

In this case the advantage is more mitigated. It is also important to note that total effect indices contain more informations on the importance of each input, thus giving greater weight to Saltelli's strategy.

ANNEXE B

Notes additionnelles

Dans cette annexe, nous nous proposons de répondre à des questions majeures soulevées par Art B. Owen. Ces questions portent sur la présence de collisions lors de l'estimation des indices de Sobol' d'ordre un ou fermés d'ordre deux à l'aide de la méthode répliquée (Section 2.3 page 47), ainsi que sur le gain (en terme de nombre d'évaluations du modèle) apporté par l'utilisation de tableaux orthogonaux. Nous commençons par traiter la question de présence de collisions.

B.1 Présence de collisions

Nous abordons d'abord l'étude de la question de présence de collisions dans le cas de l'estimation des indices d'ordre un par la méthode répliquée.

B.1.1 Cas des indices d'ordre un

Etude probabiliste

Nous rappelons que l'estimation des indices d'ordre un par la méthode répliquée se fait à l'aide de deux hypercubes latins répliqués d'ordre un (équation (2.13) page 48) notés $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ où $\mathbf{X}_i = (X_{i,1}, \dots, X_{i,d})$ et $\mathbf{X}'_i = (X'_{i,1}, \dots, X'_{i,d})$. Nous menons le raisonnement sur la structure discrète des deux hypercubes latins répliqués : $X_{i,k} = \pi_k^{(1)}(i)$ et $X'_{i,k} = \pi_k^{(2)}(i)$, $k \in \{1, \dots, d\}$.

Pour $j \in \{1, \dots, d\}$, nous rappelons que l'estimation de l'indice de Sobol' d'ordre un S_j consiste en un réarrangement des lignes de $\mathcal{P}'$ à l'aide d'une permutation π_j tel que $X_{i,j} = X'_{\pi_j(i),j}$. Le problème soulevé est le suivant : pour $k \neq j$ on peut obtenir $X_{i,k} = X'_{\pi_j(i),k}$. Nous appelons collision un tel événement. Dans [Owe91], Owen utilise la terminologie *coincidence defect* à la place de collision. Si une collision a lieu, les lignes $\mathbf{X}_i$ et $\mathbf{X}'_{\pi_j(i)}$ ont en commun plus d'une coordonnée et des effets de confusion (*confounding*) ont lieu entre S_j et des indices d'ordres supérieurs.

Il est possible de probabiliser le nombre de collisions pour un couple de lignes $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$, $i \in \{1, \dots, n\}$. Pour $k \neq j$, $k \in \{1, \dots, d\}$, $X'_{\pi_j(i),k}$ est une variable aléatoire prenant ses valeurs dans $\{1, \dots, n\}$. La probabilité qu'une collision ait lieu entre $\mathbf{X}_i$ et $\mathbf{X}'_{\pi_j(i)}$ en position k est donnée par :

$$\sum_{l=1}^n \mathbb{P} \left[X_{i,k} = X'_{\pi_j(i),k} \mid X_{i,k} = l \right] \mathbb{P} [X_{i,k} = l] = \sum_{l=1}^n \frac{1}{n} \times \frac{1}{n} = \frac{1}{n}.$$

Etant donné que les $d - 1$ variables aléatoires $\{\mathbf{X}'_{\pi_j(i),k}\}_{k=1,\dots,d,k \neq j}$ sont indépendantes, le nombre de collisions pour le couple de lignes $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ peut être modélisé par une loi binomiale de paramètres $d - 1$ et $1/n$. Notons par C la variable aléatoire modélisant le nombre de collisions :

$$P[C = l] = \binom{n}{l} \left(\frac{1}{n}\right)^l \left(1 - \frac{1}{n}\right)^{n-l}, \quad l \in \{0, \dots, d - 1\}.$$

La table B.1 ci-dessous donne quelques valeurs de $P[C \geq 1]$ (une collision ou plus) en fonction de d et n . L'étude de cette table nous permet d'intuiter le nombre

TABLE B.1 – Valeurs prises par $P[C \geq 1]$ (une collision ou plus) pour différentes valeurs de d et n .

$d \setminus n$	50	100	250	500	750	1000
5	0.077	0.039	0.016	0.008	0.005	0.004
10	0.166	0.086	0.035	0.018	0.012	0.009
15	0.246	0.131	0.055	0.028	0.019	0.014
20	0.319	0.174	0.073	0.037	0.025	0.019
25	0.384	0.214	0.092	0.047	0.032	0.024
30	0.443	0.253	0.110	0.056	0.037	0.028

potentiel de collisions pour l'ensemble des n couples de lignes $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}_{i=1}^n$ pour l'estimation d'un indice S_j . Pour $d = 30$ variables d'entrée, si l'on choisit $n = 1000$, en moyenne 28 couples de lignes parmi les 1000 donnent lieu à une collision ou plus.

Cependant, même si un nombre important de couples de lignes $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ donnent lieu à des collisions, les positions k en lesquels les collisions ont lieu diffèrent d'un couple de lignes à l'autre. Ainsi, plutôt que d'obtenir un effet de confusion important entre S_j et un indice d'ordre supérieur, on obtient une multitude de petits effets de confusion entre S_j et différents indices d'ordres supérieurs.

Nous proposons ci-dessous une solution pour assurer qu'aucune collision n'ait lieu lors de l'estimation de chaque indice S_j , $j \in \{1, \dots, d\}$.

Solution

La solution que nous proposons consiste à construire deux hypercubes latins sans collisions. Pour ce faire, nous considérons un choix particulier des permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}$ et $\pi_1^{(2)}, \dots, \pi_d^{(2)}$ engendrant les deux hypercubes latins répliqués $\mathcal{P}$ et $\mathcal{P}'$ suivant la définition 2.13. Remarquons qu'avec cette définition, pour $j \in \{1, \dots, d\}$, la permutation π_j réarrangeant les lignes de $\mathcal{P}'$ s'écrit :

$$\pi_j = \pi_j^{(2)^{-1}} \circ \pi_j^{(1)}.$$

L'idée sous-jacente à la solution que nous proposons est l'utilisation de permutations circulaires d'ordre n appelées n -cycles. Soit Π_n l'ensemble des permutations de $\{1, \dots, n\}$. Un n -cycle, noté π , est une permutation de Π_n vérifiant :

$$\begin{cases} \pi^n = \underbrace{\pi \circ \pi \circ \dots \circ \pi}_{n \text{ fois}} = I_n, \\ \pi^\alpha(i) \neq \pi^\beta(i), \quad (\alpha, \beta, i) \in \{1, \dots, n\}^3, \alpha \neq \beta, \end{cases} \quad (\text{B.1})$$

où I_n est la permutation identité et $\circ$ désigne la loi de composition dans Π_n . Il existe $(n-1)!$ n -cycles distincts dans Π_n . La construction d'un n -cycle peut se faire à l'aide de l'algorithme de Sattolo [Sat86]. Cet algorithme génère aléatoirement et de façon équiprobable un n -cycle parmi les $(n-1)!$ existants.

Une fois un n -cycle π construit, nous proposons le choix suivant :

$$\forall j \in \{1, \dots, d\}, \begin{cases} \pi_j^{(1)} = \pi^{d-j} \\ \pi_j^{(2)} = \pi^{d+j} \end{cases} \quad (\text{B.2})$$

Remarquons qu'un tel mapping implique de choisir $n > 2d$ afin d'éviter qu'une permutation $\pi_j^{(1)}$ soit confondue avec une permutation $\pi_{j'}^{(2)}$. De plus, pour un hypercube latin, la probabilité de tomber dans cette configuration sans collisions est de $\frac{(n-1)!}{(n!)^d}$ où $(n-1)!$ correspond au choix du n -cycle et $(n!)^d$ est le nombre d'hypercubes latins à d colonnes.

Il reste maintenant à montrer qu'avec un tel choix, pour chaque permutation π_j , chaque couple de lignes $\{\mathbf{X}_i, \mathbf{X}'_{\pi_j(i)}\}$ donne lieu à aucune collision. Remarquons tout d'abord que π_j se réécrit :

$$\pi_j = \pi_j^{(2)^{-1}} \circ \pi_j^{(1)} = \pi^{n-(d+j)+d-j} = \pi^{-2j}.$$

Raisonnons par l'absurde et supposons qu'il existe une collision, alors il existe $k \neq j$ tel que $X_{i,k} = X'_{\pi_j(i),k}$. Autrement dit, il existe $k \neq j$ tel que :

$$\pi_k^{(1)}(i) = \pi_k^{(2)} \circ \pi_j(i),$$

c'est à dire :

$$\pi^{d-k}(i) = \pi^{d+k-2j}(i),$$

d'après (B.1) :

$$\begin{aligned} \pi^{d-k}(i) = \pi^{d+k-2j}(i) &\Rightarrow d-k = d+k-2j, \\ &\Rightarrow k = j. \end{aligned}$$

Or $k \neq j$ d'où l'absurdité. Ainsi le choix (B.2) proposé pour les permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}$ et $\pi_1^{(2)}, \dots, \pi_d^{(2)}$ engendrant $\mathcal{P}$ et $\mathcal{P}'$ garantit qu'aucune collision n'ait lieu lors de l'estimation de chaque indice S_j .

Un travail complémentaire consistera à regarder l'influence d'un tel mapping sur le comportement de l'estimateur de S_j . D'une part l'espace dans lequel la permutation cyclique π est tirée est plus restreint. D'autre part, le mapping (B.2) ne garantit plus l'indépendance des permutations $\pi_1^{(1)}, \dots, \pi_d^{(1)}$ et $\pi_1^{(2)}, \dots, \pi_d^{(2)}$. Un travail futur sera donc de caractériser le biais engendré par une telle solution.

B.1.2 Cas des indices fermés d'ordre deux

Le cas des indices fermés d'ordre deux se traite de manière analogue au cas des indices d'ordre un. Soit $\mathcal{P} = \{\mathbf{X}_i\}_{i=1}^n$ et $\mathcal{P}' = \{\mathbf{X}'_i\}_{i=1}^n$ les deux tableaux orthogonaux randomisés de force deux et répliqués définis par (2.14) page 51. Nous rappelons que l'estimation de l'indice de Sobol' fermé d'ordre deux $\underline{S}_{\{k,l\}}$ consiste en un réarrangement des lignes de $\mathcal{P}'$ à l'aide d'une permutation $\pi_{\{k,l\}}$ tel que $X_{i,\{k,l\}} = X'_{\pi_{\{k,l\}}(i),\{k,l\}}$.

En suivant un raisonnement similaire à celui présenté dans la section B.1.1, on montre que pour un couple de ligne $\{\mathbf{X}_i, \mathbf{X}'_{\pi_{\{k,l\}}(i)}\}$, le nombre de collisions suit une loi binomiale de paramètres $d - 2$ et $1/q$. Nous notons à nouveau par C la variable aléatoire modélisant le nombre de collisions pour un couple de ligne $\{\mathbf{X}_i, \mathbf{X}'_{\pi_{\{k,l\}}(i)}\}$. La table B.2 ci-dessous donne quelques valeurs de $P[C \geq 1]$ (une collision ou plus) en fonction de d et q .

TABLE B.2 – Valeurs prises par $P[C \geq 1]$ (une collision ou plus) pour différentes valeurs de d et q .

$d \setminus q$	7	11	16	23	29	32
5	0.370	0.249	0.176	0.125	0.100	0.091
10	0.709	0.533	0.403	0.299	0.245	0.224
15	0.865	0.710	0.568	0.439	0.366	0.338
20	0.938	0.820	0.687	0.551	0.468	0.435
25	0.971	0.888	0.773	0.640	0.554	0.518
30	0.987	0.931	0.836	0.712	0.626	0.589

La probabilité d'obtenir une ou plusieurs collisions est beaucoup plus élevée dans le cas de l'estimation des indices fermés d'ordre deux. Cela s'explique par le nombre de valeurs prises par chaque variable d'entrée. Pour une taille n de $\mathcal{P}$ et $\mathcal{P}'$, dans le cas des indices d'ordre un, chaque variable d'entrée prend n valeurs contre seulement $q \approx \sqrt{n}$ dans le cas des indices fermés d'ordre deux.

Toutefois, comme expliqué dans la section précédente les collisions n'ont pas toutes lieu en une même position k . C'est pourquoi même si le nombre de collisions est élevé, en pratique on observe pas de biais prononcé dans l'estimation des indices fermés d'ordre deux.

Au contraire du cas des indices d'ordre un, trouver une solution pour réduire le nombre de collisions est beaucoup plus complexe. Cela est principalement dû au fait que les colonnes d'un tableau orthogonal ne s'expriment pas sous la forme de permutations. Réduire le nombre de collisions lors de l'utilisation de tableaux orthogonaux est une piste de recherche à explorer.

B.2 Tableaux orthogonaux et gain

Dans cette section nous nous proposons de détailler plus précisément le gain apporté par l'utilisation de tableaux orthogonaux pour l'estimation des indices fermés d'ordre deux. Rappelons que l'estimation de ces indices par la méthode répliquée se fait à l'aide de deux tableaux orthogonaux randomisés de force deux (équation 2.14).

Le coût de cette méthode d'estimation est $2q^2$ où q est le nombre de niveaux du tableau orthogonal. q doit satisfaire la contrainte de construction du tableau orthogonal : $q \geq d - 1$. Ainsi, le coût de la méthode répliquée est au minimum de $2(d - 1)^2$. Le coût d'estimation des indices fermés d'ordre deux par la méthode classique de Sobol' est $n \left(\frac{d(d - 1)}{2} + 1 \right)$. Nous comparons ci-dessous ces deux coûts pour deux configurations : d petit et d grand.

Pour des petites valeurs de d , on peut choisir $n = q^2 > (d - 1)^2$ avec suffisamment de liberté. La différence de coût, notée D , entre la méthode répliquée et la méthode classique s'écrit :

$$D = n \left(1 - \frac{d(d - 1)}{2} \right).$$

L'estimation des indices fermés d'ordre deux étant pertinente pour $d \geq 3$, on a toujours $D < 0$.

Pour des grandes valeurs de d , on est plus rapidement amené à devoir choisir $q = d - 1$. La différence de coût, notée D , entre la méthode répliquée et la méthode classique s'écrit :

$$\begin{aligned} D &= 2(d - 1)^2 - n \left(\frac{d(d - 1)}{2} + 1 \right) \\ D &= (d - 1)^2 \left(2 - \frac{n}{2} \right) - \frac{n(d + 1)}{2}. \end{aligned}$$

Cette différence est négative dès que $n > 4$ dans la méthode classique, ce qui est tout le temps le cas (choisir $n \leq 4$ revient à construire des plans d'expériences contenant au plus 4 points).

On peut également comparer la méthode répliquée pour l'estimation des indices d'ordre un et deux par rapport à la première stratégie de Saltelli (théorème 2.1). Le coût de la stratégie de Saltelli est $n(d + 2)$ pour estimer indices d'ordre un et indices

totaux. Le coût de la méthode répliquée pour estimer indices d'ordre un et d'ordre deux est $2n + 2q^2$. La différence entre ces deux coûts vaut :

$$D = 2q^2 - nd.$$

Pour des petites valeurs de d , considérons à nouveau $n = q^2$, $q > d - 1$, la différence devient :

$$D = n(2 - d),$$

ce qui est l'avantage de la méthode répliquée.

Pour des grandes valeurs de d , considérons $q = d - 1$, la différence vaut :

$$D = (d - 1)(2(d - 1) - n) - n.$$

Cette différence est à l'avantage de la méthode répliquée lorsque $n > 2(d - 1)$. Il est toutefois important de rappeler que les indices totaux contiennent plus d'informations sur l'influence des variables d'entrée que les indices fermés d'ordre deux.

ANNEXE C

Code Maple

Le code Maple présenté dans cette annexe permet de calculer les indices de Sobol' d'ordre un, les indices fermés d'ordre deux et les indices totaux d'une fonction pour laquelle on dispose d'une expression analytique et pour des lois de distribution uniformes sur $[0, 1]$ des variables d'entrée de la fonction. Nous commençons par détailler les paramètres d'entrée et de sortie du code.

Paramètres d'entrée

- f : une expression Maple de la fonction dont on cherche à calculer les indices de Sobol'.
- d : un entier précisant le nombre de variables d'entrée de la fonction.
- $order$: un entier appartenant à $\{0, 1, 2\}$ précisant la catégorie d'indices de Sobol' à calculer : 0 pour les indices totaux, 1 pour les indices d'ordre un et 2 pour les indices fermés d'ordre deux.

Paramètres de sortie

- V : un nombre réel correspondant à la moyenne $E[f]$ de la fonction f .
- m : un nombre réel correspondant à la variance $\text{Var}[f]$ de la fonction f .
- S : une liste ou un conteneur *Array* contenant l'ensemble des indices de Sobol' calculés. Si $order == 0$ ou $order == 1$, S est une liste contenant les d indices totaux ou d'ordre un. Si $order == 2$, S est un conteneur *Array* regroupant les $d(d-1)/2$ indices fermés d'ordre deux.

Code

```
> with(combinat) :  
soboltheoric := proc(f, d, order)  
    local i, k, l, m, S, M, V, fun, argmts, vinit, val1, val2, mpartiel,  
        mpartiel2, empartiel, keepset, intrange, newl, couple;  
    vinit := [0$d];  
    fun := f(op([1, 1], f));  
    argmts := [op([1, 1], f)];  
    newl := [seq(i, i = 1..d)];
```

```

for  $i$  in  $newl$  do
     $vinit[i] := argmts[i] = 0..1;$ 
od;
 $m := int(fun, map(op, map(op, newl), vinit));$ 
 $V := int(fun ** 2, map(op, map(op, newl), vinit));$ 
 $V := V + int(-2 * fun * m, map(op, map(op, newl), vinit));$ 
 $V := V + int(m ** 2, map(op, map(op, newl), vinit));$ 
if  $order = 1$  then
     $S := [0\$d];$ 
    for  $k$  from 1 to  $d$  do
         $keepset := remove(has, newl, newl[k]);$ 
         $keepset := map(op, keepset);$ 
         $intrange := map(op, keepset, vinit);$ 
         $mpartiel := int(fun, intrange);$ 
         $intrange := map(op, newl[k], vinit);$ 
         $empartiel := int(mpartiel ** 2, intrange);$ 
         $S[k] := evalf(value((empartiel - m ** 2)/V));$ 
    od
fi;
if  $order = 0$  then
     $S := [0\$d];$ 
    for  $k$  from 1 to  $d$  do
         $keepset := remove(has, newl, newl[k]);$ 
         $keepset := map(op, keepset);$ 
         $intrange := map(op, newl[k], vinit);$ 
         $mpartiel := int(fun, intrange);$ 
         $mpartiel2 := expand(mpartiel^2);$ 
         $intrange := map(op, keepset, vinit);$ 
         $empartiel := int(mpartiel2, intrange);$ 
         $S[k] := 1 - evalf(value((empartiel - m ** 2)/V));$ 
    od
fi;
if  $order = 2$  then
     $couple := choose(d, 2);$ 
     $S := Array([0\$nops(couple)]);$ 

```

```

for  $l$  from 1 to  $nops(couple)$  do
     $val1 := newl[couple[l][1]]$ ;
     $val2 := newl[couple[l][2]]$ ;
     $keepset := remove(has, newl, val1)$ ;
     $keepset := remove(has, keepset, val2)$ ;
     $keepset := map(op, keepset)$ ;
     $intrange := map(op, keepset, vinit)$ ;
     $mpartiel := int(fun, intrange)$ ;
     $intrange := map(op, [op(val1), op(val2)], vinit)$ ;
     $empartiel := int(mpartiel ** 2, intrange)$ ;
     $S[l] := evalf(value((empartiel - m ** 2)/V))$ ;
od
fi;
return  $\{m, V, S\}$ ;
end proc

```

Exemple d'utilisation

Considérons la fonction test Ishigami introduite dans [IH90b], formulée ici dans sa version recentrée sur $[0, 1]^d$:

$$f: \begin{cases} [0, 1]^d & \rightarrow \mathbb{R} \\ (x, y, z) & \mapsto \sin(2\pi x - \pi) + 7 \sin^2(2\pi y - \pi) + 0.1(2\pi z - \pi)^4 \sin(2\pi x - \pi) \end{cases}$$

Afin de calculer les indices de Sobol' d'ordre un, fermés d'ordre deux et totaux de cette fonction, on commence par donner l'expression Maple de f :

```
>  $ishigami := (x, y, z) \mapsto \sin(2 * Pi * x - Pi) + 7 * \sin(2 * Pi * y - Pi) ** 2 + 0.1 * (2 * Pi * z - Pi) ** 4 * \sin(2 * Pi * x - Pi)$ ;
```

Le calcul des indices d'ordre un est obtenu par la commande :

```
>  $first\_ishigami := soboltheoric(ishigami, 3, 1)$ ;
 $first\_ishigami := \{3.500000000, 13.84, [0.3139, 0.4424, 0.0]\}$ 
```

Le calcul des indices fermés d'ordre deux est obtenu par la commande :

```
>  $second\_ishigami := soboltheoric(ishigami, 3, 2)$ ;
 $first\_ishigami := \{3.5, 13.84, [0.7563, 0.5575, 0.4424]\}$ 
```

Le calcul des indices totaux est obtenu par la commande :

```
>  $total\_ishigami := soboltheoric(ishigami, 3, 0)$ ;
 $total\_ishigami := \{3.5, 13.84, [0.5575, 0.4424, 0.2436]\}$ 
```


Bibliographie

- [AH00] J. E. Abraham and J. D. Hunt. Parameter estimation strategies for large-scale urban models. *Transportation Research Record : Journal of the Transportation Research Board*, 1722(1) :9–16, 2000.
- [BB52] R. C. Bose and K. A. Bush. Orthogonal Arrays of strength two and three. *Ann. Math. Stat.*, 23(4) :508–524, 1952.
- [BD87] G. E. P. Box and N. R. Draper. *Empirical Model Building and Response Surfaces*. John Wiley & Sons, New York, NY, 1987.
- [BFN92] P. Bratley, B. L. Fox, and H. Niederreiter. Implementation and tests of low-discrepancy sequences. *ACM Trans. Model. Comput. Simul.*, 2(3) :195–213, 1992.
- [BHH78] G. E. P. Box, W. G. Hunter, and J. S. Hunter. *Statistics for Experimenters : An Introduction to Design, Data Analysis and Model Building*. Wiley, New York, NY, 1978.
- [Bus52] K. A. Bush. Orthogonal Arrays of index unity. *Ann. Math. Statist.*, 23(3) :426–434, 1952.
- [CA14] B. Ciuffo and C. L. Azevedo. A sensitivity-analysis-based approach for the calibration of traffic simulation models. *IEEE Trans. Intell. Transp. Syst.*, 15(3) :1298–1309, 2014.
- [Can12] Y. Caniou. *Analyse de sensibilité globale pour les modèles imbriqués et multi-échelles*. PhD thesis, Laboratoire de Mécanique et Ingénieries (LaMI), Université Blaise Pascal et Institut Français de Mécanique Avancée, 2012.
- [CFS⁺73] R. I. Cukier, C. M. Fortuin, K. E. Shuler, A. G. Petschek, and J. H. Schaibly. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I Theory. *J. Chem. Phys.*, 59(8) :3873, 1973.
- [Cha13] G. Chastaing. *Indices de Sobol’ généralisés pour variables dépendantes*. PhD thesis, Laboratoire Jean Kuntzmann, Université Grenoble Alpes, France, 2013.
- [CLS78] R. I. Cukier, H. B. Levine, and K. E. Shuler. Nonlinear sensitivity analysis of multiparameter model systems. *J. Comput. Phys.*, 26(1) :1–42, 1978.
- [CMO97] R. E. Caffisch, W. J. Morokoff, and A. B. Owen. Valuation of mortgage backed securities using brownian bridges to reduce effective dimension. *J. Comput. Finance*, 1(1) :27–46, 1997.
- [CP76] R. Cranley and T. N. L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM J. Numer. Anal.*, 13(6) :904–914, 1976.

- [CSS75] R. I. Cukier, J. H. Schaibly, and K. E. Shuler. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. III. Analysis of the approximations. *J. Chem. Phys.*, 63(3) :1140, 1975.
- [CSV15] T. Capelle, P. Sturm, and A. Vidard. Formulating LUTI calibration as an optimisation problem : example of Tranus shadow price estimation. *Procedia Engineering*, 115 :12–20, 2015.
- [DBH14] W. Daamen, C. Buisson, and S. P. Hoogendoorn. *Traffic Simulation and Data : Validation Methods and Applications*. Taylor & Francis Group, 2014.
- [dC35] J. Van der Corput. Verteilungsfunktionen. I. Mitt. *Proc. Akad. Wet. Amsterdam*, 38 :813–821, 1935.
- [DCI13] G. Damblin, M. Couplet, and B. Iooss. Numerical studies of space-filling designs : optimization of Latin Hypercube Samples and subprojection properties. *Journal of Simulation*, 7(4) :276–289, 2013.
- [Dev86] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, New York, 1986.
- [Dey12] A. Dey. On the construction of nested Orthogonal Arrays. *Australas. J. Combin.*, 54 :37–48, 2012.
- [dlB99] T. de la Barra. Mathematical description of TRANUS. Technical report, Modelistica, 1999. <http://www.tranus.com/tranus-english>.
- [DSA⁺12] P. Dutta, M. Saujot, E. Arnaud, B. Lefevre, and E. Prados. Uncertainty propagation and sensitivity analysis during calibration of TRANUS, an integrated land use and transport model. In *International Conference on Urban, Regional Planning and Transportation*, 2012.
- [Efr81] B. Efron. Nonparametric standard errors and confidence intervals. *Can. J. Stat.*, 9(2) :139–158, 1981.
- [ET93] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman & Hall/CRC, 1993.
- [Fau82] H. Faure. Discrepance de suites associées à un système de numération (en dimension s). *Acta Arith.*, 41(4) :337–352, 1982.
- [Fis26] R. A. Fisher. The arrangement of field experiments. *J. Ministry Agric.*, 33 :503–513, 1926.
- [FLS06] K.-T. Fang, R. Li, and A. Sudjianto. *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC, Computer Science and Data Analysis Series edition, 2006.
- [Fra08] J. Franco. *Planification d’expériences numériques en phase exploratoire pour la simulation des phénomènes complexes*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne, France, 2008.
- [FVCS09] J. Franco, O. Vasseur, B. Corre, and M. Sergent. Minimum Spanning Tree : A new approach to assess the quality of the design of computer experiments. *Chemom. Intell. Lab. Syst.*, 97(2) :164–169, 2009.

- [GACP16] L. Gilquin, E. Arnaud, T. Capelle, and C. Prieur. Sensitivity analysis and optimisation of a Land Use and Transport Integrated model. preprint available at <https://hal.inria.fr/hal-01291774>, 2016.
- [GAMP16] L. Gilquin, E. Arnaud, H. Monod, and C. Prieur. Recursive estimation procedure of Sobol’ indices based on replicated designs. preprint available at <https://hal.inria.fr/hal-01291769>, 2016.
- [GAP15] L. Gilquin, E. Arnaud, and C. Prieur. Replication procedure for grouped Sobol’ indices estimation in dependent uncertainty spaces. *Inf. Inference*, 4(4) :354–379, 2015.
- [GCM14] Q. Ge, B. Ciuffo, and M. Menendez. An exploratory study of two efficient approaches for the sensitivity analysis of computationally expensive traffic simulation models. *IEEE Trans. Intell. Transp. Syst.*, 15(3) :1288–1297, 2014.
- [GJKL14] F. Gamboa, A. Janon, T. Klein, and A. Lagnoux. Sensitivity analysis for multidimensional and functional outputs. *Electron. J. Statist.*, 8(1) :575–603, 2014.
- [GM14] Q. Ge and M. Menendez. An efficient sensitivity analysis approach for computationally expensive microscopic traffic simulation models. *International Journal of Transportation*, 2(2) :49–64, 2014.
- [GR16] L. Gilquin and Ll. A. Jiménez Rugama. Reliable error estimation for Sobol’ indices. draft version available at <https://hal.inria.fr/hal-01358067v1>, 2016.
- [GRA⁺16] L. Gilquin, Ll. A. Jiménez Rugama, E. Arnaud, F. J. Hickernell, H. Monod, and C. Prieur. Iterative construction of replicated designs based on Sobol’ sequences. preprint available at <https://hal.inria.fr/hal-01349444>, 2016.
- [GS03] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements : A Spectral Approach*. Springer New York, 2003.
- [HH03] H. S. Hong and F. J. Hickernell. Algorithm 823 : Implementing scrambled digital sequences. *ACM Trans. Math. Software*, 29(2) :95–109, 2003.
- [HI86] S. C. Hora and R. L. Iman. A comparison of maximum/bounding and Bayesian/Monte Carlo for fault tree uncertainty analysis. Technical report, SAND85-2839, Sandia National Laboratories, Albuquerque, NM, 1986.
- [Hic98a] F. J. Hickernell. A generalized discrepancy and quadrature error bound. *Math. Comp.*, 67(221) :299–322, 1998.
- [Hic98b] F. J. Hickernell. Lattice rules : How well do they measure up? In P. Hellekalek and G. Larcher, editors, *Random and Quasi-Random Point Sets*, pages 109–166. Springer New York, 1998.

- [Hla61] E. Hlawka. Funktionen von beschränkter variation in der theorie der gleichverteilung. *Annali di Matematica Pura ed Applicata*, 54(1) :325–333, 1961.
- [Hoe48] W. F. Hoeffding. A class of statistics with asymptotically normal distribution. *Ann. Math. Stat.*, 19(3) :293–325, 1948.
- [HR16] F. J. Hickernell and Ll. A. Jiménez Rugama. Reliable adaptative cubature using digital sequences. *Monte Carlo and Quasi-Monte Carlo Methods*, 163 :367–383, 2016.
- [HS96] T. Homma and A. Saltelli. Importance measures in global sensitivity analysis of nonlinear models. *Reliab. Eng. Syst. Saf.*, 52(1) :1–17, 1996.
- [HSS99] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal Arrays : Theory and Applications*. New York : Springer, springer ser. statist. edition, 1999.
- [HT13] Y. He and B. Tang. Strong orthogonal arrays and associated Latin hypercubes for computer experiments. *Biometrika*, 100(1) :254–260, 2013.
- [HT14] Y. He and B. Tang. A characterization of strong orthogonal arrays of strength three. *Ann. Statist.*, 42(4) :1347–1360, 2014.
- [IC82] R. L. Iman and W. J. Conover. A distribution-free approach to inducing rank correlation among input variables. *Comm. Statist. Simulation Comput.*, 11(3) :311–334, 1982.
- [IH90a] R. L. Iman and S. C. Hora. A robust measure of uncertainty importance for use in fault tree system analysis. *Risk Anal.*, 10(3) :401–406, 1990.
- [IH90b] T. Ishigami and T. Homma. An importance quantification technique in uncertainty analysis for computer models. In *Uncertainty Modeling and Analysis, 1990. Proceedings., First International Symposium on*, pages 398–403, 1990.
- [Jan08] A. Janon. *Analyse de sensibilité et réduction de dimension. Application à l’océanographie*. PhD thesis, Laboratoire Jean Kuntzmann, Université Grenoble Alpes, France, 2008.
- [JK08] S. Joe and F. Y. Kuo. Constructing Sobol’ sequences with better two-dimensional projections. *SIAM J. Sci. Comput.*, 30(5) :2635–2654, 2008.
- [JKL⁺14] A. Janon, T. Klein, A. Lagnoux, M. Nodet, and C. Prieur. Asymptotic normality and efficiency of two Sobol’ index estimators. *ESAIM Probab. Stat.*, 18 :342–364, 2014.
- [JLD06] J. Jacques, C. Lavergne, and N. Devictor. Sensitivity analysis in presence of model uncertainty and correlated inputs. *Reliab. Eng. Syst. Saf.*, 91(10-11) :1126–1134, 2006.
- [JMY90] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *J. Statist. Plann. Inference*, 26(2) :131–148, 1990.

- [JSW98] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *J. Global Optim.*, 13 :455–492, 1998.
- [Kis42] K. Kishen. On Latin and hyper-Graeco-Latin cubes and Hypercubes. *Current Science*, 11(3) :98–99, 1942.
- [KS67] M. G. Kendall and A. Stuart. *The Advanced Theory of Statistics, Volume 2 : Inference and Relationship*. Griffin, 1967.
- [Lem09] C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. New York : Springer, springer ser. statist. edition, 2009.
- [Leo41] W. W. Leontief. *The Structure of the American Economy 1919-1939*. New York : Oxford University Press, 2nd edition, 1951 edition, 1941.
- [LMM11] M. Lamboni, H. Monod, and D. Makowski. Multivariate sensitivity analysis to measure global contribution of input factors in dynamic models. *Reliab. Eng. Syst. Saf.*, 96(4) :450–459, 2011.
- [Loh96] W. L. Loh. On Latin Hypercube Sampling. *Ann. Statist.*, 24(5) :2058–2080, 1996.
- [Loh08] W. L. Loh. A multivariate central limit theorem for randomized Orthogonal Array sampling designs in computer experiments. *Ann. Statist.*, 36(4) :1983–2023, 2008.
- [LR11] G. Li and H. Rabitz. General formulation of HDMR component functions with independent and correlated variables. *J. Math. Chem.*, 50(1) :99–130, 2011.
- [LS63] W. W. Leontief and A. Strout. *Multi-Regional Input-Output Analysis*. Structural Interdependence and Economic Development. London : Mcmillan, 1963.
- [Mat98] J. Matoušek. On the L2-discrepancy for anchored boxes. *J. Complexity*, 14(4) :527–556, 1998.
- [Mat99] J. Matoušek. *Geometric Discrepancy*. Springer, Berlin, algorithms and combinatorics edition, 1999.
- [MBC79] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2) :239–245, 1979.
- [MC94] W. J. Morokoff and R. E. Caflisch. Quasi-random sequences and their discrepancies. *SIAM J. Sci. Comput.*, 15(6) :1251–1279, 1994.
- [McF73] D. McFadden. Conditional logit analysis of qualitative choice behaviour. *Frontiers in Econometrics - Academic Press New York*, pages 105–142, 1973.
- [McK95] M. D. McKay. Evaluating prediction uncertainty. Technical report, Los Alamos National Laboratory Report NUREG/CR- 6311, LA-12915-MS., 1995.

- [MJ08] T. A. Mara and O. Rakoto Joseph. Comparison of some efficient methods to evaluate the main effect of computer model factors. *J. Statist. Comput. Simulation*, 78(2) :167–178, 2008.
- [MM95] R. H. Myers and D. C. Montgomery. *Response Surface Methodology : Process and Product Optimization Using Designed Experiments*. Wiley, New York, NY, 1995.
- [MMM06] M. D. Morris, L. M. Moore, and M. D. McKay. Sampling plans based on balanced incomplete block designs for evaluating the importance of computer model inputs. *J. Statist. Plann. Inference*, 136(9) :3203–3220, 2006.
- [MMM08] M. D. Morris, L. Moore, and M. D. McKay. Orthogonal Arrays in the sensitivity analysis of computer models. *Technometrics*, 50(2) :205–215, 2008.
- [MNM06] H. Monod, C. Naud, and D. Makowski. *Uncertainty and sensitivity analysis for crop models*, chapter 3, pages 55–100. Elsevier, 2006.
- [Mor91] M. D. Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2) :161–174, 1991.
- [MT00] D. McFadden and K. Train. Mixed MNL models for discrete response. *Journal of Applied Econometrics*, 15 :447–470, 2000.
- [MTZ78] J. Mockus, V. Tiesis, and A. Zilinskas. The application of bayesian methods for seeking the extremum. In L. C. W. Dixon and G. P. Szego, editors, *Towards Global Optimisation, Vol. 2*, pages 117–129. North-Holand, 1978.
- [Nie92a] H. Niederreiter. Low-discrepancy point sets obtained by digital constructions over finite fields. *Czechoslovak Mathematical Journal*, 42(1) :143–166, 1992.
- [Nie92b] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. SIAM, Philadelphia, pacbms-nsf regional conference series in applied math. edition, 1992.
- [NX96] H. Niederreiter and C. Xing. Low-discrepancy sequences and global function fields with many rational places. *Finite Fields Appl.*, 2(3) :241–273, 1996.
- [OW11] J. D. Ortúzar and L. G. Willumsen. *Modelling Transport*. Wiley, Fourth Edition edition, 2011.
- [Owe91] A. B. Owen. Lattice sampling revisited : Monte carlo variance of means over randomized orthogonal arrays. *Ann. Statist.*, 22(2) :930–945, 1991.
- [Owe92] A. B. Owen. Orthogonal Arrays for computer experiments, integration and visualization. *Statist. Sinica*, 2(2) :439–452, 1992.
- [Owe94] A. B. Owen. Lattice sampling revisited : Monte Carlo variance of means over randomized orthogonal arrays. *Ann. Statist.*, 22(2) :930–945, 1994.

- [Owe95] A. B. Owen. Randomly permuted (t, m, s) -nets and (t, s) -sequences. In H. Niederreiter and P. J. S. Shiue, editors, *Monte Carlo and Quasi-Monte Carlo methods in scientific computing*, pages 299–317. Springer-Verlag, New York, 1995.
- [Owe97a] A. B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM J. Numer. Anal.*, 34(5) :1884–1910, 1997.
- [Owe97b] A. B. Owen. Scrambled net variance for integrals of smooth functions. *Ann. Statist.*, 25(4) :1541–1562, 1997.
- [Owe98] A. B. Owen. Scrambling Sobol’ and Niederreiter-Xing points. *J. Complexity*, 14(4) :466–489, 1998.
- [Owe03] A. B. Owen. Variance and discrepancy with alternative scramblings. *ACM Trans. Model. Comput. Simul.*, 13(4) :363–378, 2003.
- [Owe12] A. B. Owen. Variance components and generalized Sobol’ indices. preprint available at <http://statweb.stanford.edu/~owen/reports/sobolmatrix.pdf>, 2012.
- [Owe13a] A. B. Owen. Better estimation of small Sobol’ sensitivity indices. *ACM Trans. Model. Comput. Simul.*, 23(2) : :11, 2013.
- [Owe13b] A. B. Owen. Monte Carlo theory, methods and examples. preprint available at <http://statweb.stanford.edu/~owen/mc/Ch-var-adv.pdf>, 2013.
- [PC04] T. Pillards and R. Cools. A theoretical view on transforming low-discrepancy sequences from a cube to a simplex. *Monte Carlo Meth. and Appl.*, 10(3-4) :511–529, 2004.
- [Pei78] C. S. Peirce. The doctrine of chances. *Popular Science Monthly*, 12 :604–615, 1878.
- [PM12] L. Pronzato and W. G. Müller. Design of computer experiments : space filling and beyond. *Stat. Comput.*, 22(3) :681–701, 2012.
- [QAW09] P. Z. G. Qian, M. Ai, and C. F. J. Wu. Construction of nested space-filling designs. *Ann. Stat.*, 37(6A) :3616–3643, 2009.
- [Qia09] P. Z. G. Qian. Nested Latin Hypercube designs. *Biometrika*, 96(4) :957–970, 2009.
- [QTW09] P. Z. G. Qian, B. Tang, and C. F. J. Wu. Nested space-filling designs for computer experiments with two levels of accuracy. *Stat. Sinica*, 19 :287–300, 2009.
- [Rao46] C. R. Rao. Hypercubes of strength "d" leading to confounded designs in factorial experiments. *Bull. Calcutta Math. Soc.*, 38(3) :67–78, 1946.
- [Rao47] C. R. Rao. Factorial experiments derivable from combinatorial arrangements of arrays. *J. Royal Statist. Soc.*, 9(1) :128–139, 1947.

- [RGD12] O. Roustant, D. Ginsbourger, and Y. Deville. Dicekriging, Diceoptim : Two R packages for the analysis of computer experiments by Kriging-based metamodeling and optimization. *J. Stat. Softw.*, 51(1), 2012.
- [RH16] Ll. A. Jiménez Rugama and F. J. Hickernell. Adaptive multidimensional integration based on rank-1 lattices. *Monte Carlo and Quasi-Monte Carlo Methods*, 163 :407–422, 2016.
- [Sal02] A. Saltelli. Making best use of models evaluations to compute sensitivity indices. *Comput. Phys. Commun.*, 145(2) :280–297, 2002.
- [Sap75] G. Saporta. Dépendance et codages de deux variables aléatoires. *Revue de Statistique Appliquée*, 23(1) :43–63, 1975.
- [Sat86] S. Sattolo. An algorithm to generate a random cyclic permutation. *Information Processing Letters*, 22(6) :315–317, 1986.
- [SCS08] A. Saltelli, K. Chan, and E. M. Scott. *Sensitivity Analysis*. Wiley, 2008.
- [SM95] D. R. Stinson and J. L. Massey. An infinite class of counterexamples to a conjecture concerning nonlinear resilient functions. *J. Cryptology*, 8(3) :167–173, 1995.
- [SM08] I. M. Sobol’ and E. E. Myshetskaya. Monte Carlo estimators for small sensitivity indices. *Monte Carlo Meth. and Appl.*, 13(5-6) :455–465, 2008.
- [Sob67] I. M. Sobol’. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. Math. Phys.*, 7(4) :86–112, 1967.
- [Sob93] I. M. Sobol’. Sensitivity indices for nonlinear mathematical models. *Mathematical Modeling and Computational Experiment*, 1 :407–414, 1993.
- [Ste87] M. Stein. Large sample properties of simulations using Latin Hypercube sampling. *Technometrics*, 29(2) :143–151, 1987.
- [Tan93] B. Tang. Orthogonal Array-based Latin Hypercubes. *J. Amer. Statistic. Assoc.*, 88(424) :1392–1397, 1993.
- [TGM06] S. Tarantola, D. Gatelli, and T. A. Mara. Random balance designs for the estimation of first order global sensitivity indices. *Reliab. Eng. Syst. Saf.*, 91(6) :717–727, 2006.
- [Thi04] E. Thiémard. An algorithm to compute bounds for the star discrepancy. *J. Complexity*, 17(4) :850–880, 2004.
- [Tis12] J. Y. Tissot. *Sur la décomposition ANOVA et l’estimation des indices de Sobol’*. Application à un modèle d’écosystème marin. PhD thesis, Laboratoire Jean Kuntzmann, Université Grenoble Alpes, France, 2012.
- [Ton10] C. Tong. Self-validated variance-based methods for sensitivity analysis of model outputs. *Reliab. Eng. Syst. Saf.*, 95(3) :301–309, 2010.

-
- [TP15] J. Y. Tissot and C. Prieur. A randomized Orthogonal Array-based procedure for the estimation of first- and second-order Sobol' indices. *J. Statist. Comput. Simulation*, 85(7) :1358–1381, 2015.
- [WF03] X. Wang and K.-T. Fang. The effective dimension and quasi-Monte Carlo integration. *J. Complexity*, 19(2) :101–124, 2003.
- [Wil81] A. G. Wilson. *Optimization in Locational and Transport Analysis*. John Wiley & Sons, New York, 1981.
- [Ye98] K. Q. Ye. Orthogonal column Latin Hypercubes and their application in computer experiments. *J. Amer. Statistic. Assoc.*, 93(444) :1430–1439, 1998.

