

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université Blida 1

Faculté des Sciences
Département de Mathématiques



Mémoire de Fin d'Etude

En vue de l'obtention du diplôme de

Master En Mathématiques

Spécialité : Modélisation Stochastique et Statistique

Thème

ETUDE DE L'IMPACT DE DEUX APPROCHES DE L'ALGORITHME

DE METROPOLIS HASTING SUR LA QUALITE D'UN PLAN

NUMERIQUES GENERE SELON UN PROCESSUS PONCTUEL

Par

CHERIFI Rachid et RAHMANI Ayoub

Devant le jury composé de :

Président

Examineur

H. Elmoassaoui Maître de conférences, U. de Blida 1

Rapporteur

Remerciements

Nous tenons d'abord à remercier ALLAH qui nous a aidé et nous a donné la patience et le courage durant ces longues années d'étude.

Nous tenons à exprimer notre gratitude et nos remerciements à notre encadreur Dr : ***ElMossaoui Hichem*** d'avoir proposé et encadré ce sujet et pour ces bon explications qui nous ont éclairées le chemin de la recherche et sa collaboration avec nous dans l'accomplissement de ce modeste travail

Egalement, nous remercions l'ensemble des professeurs et enseignants de département de Mathématique de l'université Blida1, qui ont participés à cette formation

Nous n'oublions pas de remercier nos familles pour leurs prières, leur soutien et leur mot d'encouragement depuis le début de cette année, mais également pendant tout notre cursus universitaire et enfin nous remercions nos amis et nos collègues et toutes les personnes qui auront contribuées de près ou de loin à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Dédicace

Je dédie ce mémoire de fin d'étude

A mes très chers parents,

A mon **père**, qui m'a aidé à construire l'homme et la personne que je suis aujourd'hui, sans lui toute ces années n'auraient jamais pu aboutir, grâce à son soutien moral et l'éducation qu'il m'a donné, il m'a appris de ne jamais abandonner et de me battre coute que coute.

A ma **mère**, l'incarnation de la douceur et de la bienveillance, celle qui m'a appris à aimer le travail, aimer les autres et aimer la vie, dans les moments difficiles tu as été la lumière qui m'a guidé et que j'ai suivie sans me poser de questions, je te dédis toute ces années de dur labeur et ce mémoire que j'ai façonné à ton image.

A mon **frère** et à ma **sœur**, pour leurs encouragements permanents, et leur soutien moral.

Mes amis, qui m'ont accompagnés à chaque pas que je marchais, leurs réconfort inconditionnel et leur bonnes idées.

A tous mes camarades de l'université et du département de Mathématique ainsi qu'aux professeurs qui m'ont enseignés tout au long de mon parcours Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible,

Merci d'être toujours là pour moi.

Cherifi Rachid

RESUME

La diversité des plans d'expériences proposés dans la littérature vient du fait qu'il n'existe pas un plan parfait où tous les critères d'optimalité sont vérifiés en même temps. Chaque plan présente des avantages pour un certain critère d'optimalité et un inconvénient pour un autre. Il s'agit donc de trouver le meilleur compromis pour les besoins de chaque étude. Dès fois, pour générer de tels plans nous pouvons utiliser les techniques de simulation par Chaîne de Markov (MCMC), et plus précisément l'algorithme de Métropolis- Hasting

Dans ce mémoire nous étudions l'impact, d'utiliser l'approche standard de l'algorithme de Métropolis- Hasting et son approche à retard rejeté sur la génération des plans d'expériences numérique selon le modèle stochastique ponctuel de Strauss Marqué. Enfin, nous présentons une application en Matlab que nous avons réalisé afin de pouvoir comparer ces deux approches.

Mots clés : Plan d'expériences numérique, Monte Carlo par chaîne de Markov (MCMC), Métropolis-Hasting, processus ponctuels marqués.

ملخص

يأتي تنوع تصميم التجارب المقترحة في الأدبيات من حقيقة انه لا يوجد شيء مثل التصميم المثالي الذي يحقق جميع المعايير المثالية في نفس الوقت. كل خطة لها مزايا لمعيار أمثل معين وكذلك عيوب لمعيار آخر. لذلك سيكون الامر يتعلق بإيجاد حل وسط لاحتياجات في كل دراسة. في بعض الاحيان، لأنشاء مثل هذه الخطط يمكننا استخدام تقنيات المحاكاة لسلسلة ماركوف (MCMC) وبشكل ادق خوارزمية Metropolis-Hasting

في هذه الاطروحة ندرس تأثير استخدام النهج القياس لخوارزمية Metropolis-Hasting ونهج التأخير المرفوض (Retard-Rejeté) على توليد التصاميم الرقمية للتجارب وفقا لنموذج العشوائي النقطي لـ (Strauss Marque). وأخيرا نقدم تطبيقا في (Matlab) حتى تتمكن من مقارنة هذين النهجين.

الكلمات المفتاحية: تصميم تجارب الرقمية ، مونت كارلو من سلسلة ماركوف MCMC ، متروبوليسهاستينغ ، عمليات النقاط المحددة .

Abstract

The diversity of the design of experiments proposed in the literatures comes from the fact that there is not a perfect design where all the optimal criteria are verified at the same time. Each plan has advantages for some optimal criteria and a disadvantage for another. For that reason it will be better to find a compromise for the needs of each study. Sometimes, to generate such plans we will use Markov Chain simulation techniques (MCMC), and more precisely the Metropolis –Hasting algorithm.

This thesis presents the study of the impact of using the standard approach on the generation of computer designs of experiments according to the point stochastic model of Strauss Marked.

Finally, we present an application in Matlab that we realized in order to be able to compare these two approaches.

Keywords: Computer Design, Monte Carlo by Markov Chain (MCMC), Metropolis-Hasting Algorithm, marked point processes.

Dédicace

.Je dédié ce modeste travail

A mes très chers parents

*Aucune dédicace ne saurait exprimée mon respect, mon amour éternel et ma
considération pour les sacrifices que vous avez consenti pour mon instruction et
mon bien être*

A mes sœurs pour leur soutien au long de cette année

*A mon frère Mohamed Menouar pour son grand aide et effort durant cette
année*

A mon neveu et mes nièces

A mes Amis lesquels j'ai partagé le bon et le mal.

A mon binôme, ami et frère Rachid

A tout la famille RAHMANI et HADJ SADOUK

Rahmani Ayoub

TABLE DES MATIERES

RESUME	
REMERCIEMENTS	
TABLE DES MATIERES	
LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX	
INTRODUCTION	6
1. GENERALITES SUR LA METHODOLOGIE DE LA RECHERCH EXPERIMENTALE	8
1.1. Notion d'espace expérimentale	8
1.2. Notion de Modalisation mathématique	10
1.3. Quelques notions sur les plans d'expériences usuels	11
1.4. Les plans d'expériences numériques	14
2. GENERALITES SUR LES PROCESSUS PONCTUELS	25
2.1. Premières définitions et Notations	25
2.2. loi d'un processus ponctuel	26
2.3. Processus ponctuels Marqué	28
2.4. Processus ponctuels de référence	29
2.5. Processus ponctuels de Gibbs	34
2.6. Processus ponctuel de Markov	36
2.7. Quelques propriétés générales des processus ponctuel	37
3. METHODE DE SIMULATION MONTE CARLO PAR CHAINE DE MARKOV	39
3.1 Méthodes de bas de simulation	58
3.2. Chaines des Markov	42
3.3. Les méthodes de Monté Carlo par chaîne de Markov	45
4. ETUDE COMPARATIVE ENTRE L'ALGORITHME DE METROPOLIS HASTING GENERALISE ET A RETARD REJETE	47
4.1. Méthode d'échantillonnage	47
4.2. Algorithme de Metropolis Hasting généralisé	47
4.3. Algorithme de Metropolis Hasting avec retard rejeté	50
4.4. Implémentation et Comparaison	51
CONCLUSION	59
APPENDICE	60
REFERENCES BIBLIOGRAPHIE	67

	consistué de tousLes valeurs compris entre les niveaux bas et les niveaux hauts	
Figure 1.2	Chaque facteur est représenté par un axe gradué et orienté. LesD axes des facteurs sont orthogonaux entre eux .l'espace ainsi définie est l'espace expérimentale	10
Figure 1.3	Les points expérimentaux sont disposés dans le domaine d'étude défini par l'expérimentateur	12
Figure 1.4	Meilleurs emplacements des points expérimentaux	12
Figure 1.5	Plan de Box-Bahnken pour 3 facteurs	13
Figure 1.6	Plans LHS avec 5 et 10 points	15
Figure 1.7	Un plan généré par un tableau orthogonal OA_1 (25, 5, 5,2) dont les points sont centrés et projetés sur le sous-espace (X_1, X_2)	16
Figure 1.8	plan généré par un tableau orthogonal $1 OA_1$ (25, 5,5, 2) dont les points sont randomisés et projetés sur le sous-espace (X_1, X_2)	17
Figure 1.9	Un hypercube latin généré par un tableau orthogonal OA_1 (25, 5, 5,2) dont les points sont randomisés et projetés sur le sous-espace (X_1, X_2)	18
Figure 1.10	Les 13 premiers points de la suite de van der Corput en base 3	20
Figure 1.11	Les 100, 1 000 et 10 000 premiers points d'une suite de Halton en bases 2 et 3	20
Figure 1.12	(De la gauche vers la droite) Les 50, 250 et 500 premiers points d'une suite de Halton en bases 2 et 3	21
Figure 1.13	Des mauvaises projections en 2D. A gauche : les 80 premiers points d'une suite de Halton en bases 11 et 13. A droite : les 100 premiers points d'une suite de Halton en bases 17 et 19	21
Figure 1.14	(De la gauche vers la droite) Les 50, 250 et 500 premiers points d'une suite de Hammersley en base 2	22
Figure 1.15	Les 50, 250 et 500 premiers points d'une suite de Faure en dimension 2	23
Figure 2.1	Processus poisson sur une image 100×100 a l'intensité $\lambda=0,01b$, d'intensité c $\lambda=0,02 c$ processus non homogène	31
Figure 2.2	Quatre réalisations d'un processus ponctuel de Thomas sur $[0, 1]^2$ pour différents paramètre	34
Figure 4.1	Boxplots de critère de qualité R, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions.	54
Figure 4.2	Boxplots de critère de qualité de distance, calculé sur 40 plans à 30 points en dimension 3, à 40 points pour dimension 5 et à 50 points pour dimension 7.	55
Figure 4.3	Boxplots de critère de discrèpance, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions	56
Figure 4.4	Boxplots de critère de qualité recouvrement, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions.	57

Introduction

La simulation numérique est devenue un outil incontournable en recherche et développement pour évaluer par exemple des modèles d'expansion de l'univers, ou pour approcher des solutions optimales d'un stockage de déchets radioactifs ou d'une architecture d'un barrage. Les codes sont constitués de modèles numériques reliant des variables, descriptives de l'état du système simulé, à un certain nombre de paramètres caractérisant, par exemple, les conditions initiales du système ou bien certaines lois ou relations utilisées dans la modélisation. Le simulateur prend ces paramètres en entrée et délivre en sortie les réponses sur les variables d'état du système. Une simulation consiste à fixer un ou plusieurs jeux de variables d'entrée, à réaliser les calculs, puis à analyser les réponses fournies par le simulateur. Malheureusement, les difficultés essentielles proviennent du coût de calcul du simulateur et de la dimension du problème à traiter. Même après une diminution drastique de la dimension, il devient impossible pour certaines applications d'envisager d'utiliser directement le simulateur. L'idée est de remplacer le simulateur par une ou plusieurs fonctions approchées. Ce sont en général des fonctions relativement simples qui sont construites à l'aide de méthodes d'approximation ou d'interpolation et à partir des plans d'expériences numériques.

Parmi les idées utilisées pour générer de tels plans d'expériences numériques, on peut utiliser les techniques de simulation par Chaîne de Markov (MCMC) et l'algorithme de Métropolis Hasting. Dans ce contexte, on peut citer les travaux de Franco J. [1] et ceux de Elmoosaoui H.*et al.* [2, 3]. Il existe plusieurs sous-catégories d'algorithme de Metropolis-Hastings [4]. Cependant, l'algorithme de Metropolis-Hastings standard ne fonctionne généralement pas en haute dimensions, car il conduit à des prélèvements répétés très fréquents. Afin de pallier cette lacune, on peut utiliser une autre variation de l'algorithme, appelée l'algorithme de Metropolis-Hastings avec rejet retardé (MHDR) qui a été introduit par Mira en 1999 [5]. L'idée clé derrière l'algorithme de Metropolis-Hastings avec rejet retardé est de réduire la corrélation entre les états de la chaîne de Markov.

L'objectif principal de notre travail est d'étudier et comparer l'utilisation des deux versions de l'algorithme de Métropolis Hasting (L'algorithme de Métropolis Hasting sous sa forme général et l'algorithme de Metropolis-Hastings avec rejet retardé) sur la génération de plans d'expériences numériques fondés sur la théorie de processus stochastiques, particulièrement les processus ponctuels de Strauss. Ainsi, les expériences de ces plans doivent remplir au mieux le domaine expérimental afin d'obtenir des informations, notamment pour détecter les éventuelles irrégularités. On cherche donc un plan dont les points seraient le plus uniformément répartis dans l'hyper-cube unité.

Ce mémoire est composé de quatre chapitres organisés comme suit:

Le premier chapitre, présente la théorie de la méthodologie des plans d'expériences classiques et un état de l'art de plans d'expériences couramment utilisés en expérimentation numérique. Ainsi que certains critères d'optimalités pour étudier la qualité de ces plans.

Dans le deuxième chapitre, nous rappelons, dans un premier temps quelques notions générales sur les processus stochastiques ponctuels. Nous introduisons ensuite les processus ponctuels de Poisson, à partir desquels nous définissons une variété de processus ponctuels. Enfin, nous présentons les processus ponctuel de Markov qui incluent la notion d'interactions entre les objets.

Dans le troisième chapitre, la méthode de simulation des processus ponctuels est décrite sous la forme d'une chaîne de Markov, dont les concepts de base et les conditions de convergence sont présentés.

Le chapitre quatre est consacré à la représentation des deux algorithmes de Metropolis-Hastings : L'algorithme de Métropolis-Hasting sous sa forme général et l'algorithme de Metropolis-Hastings avec rejet retardé. Une comparaison entre ces deux algorithmes a été effectuée pour les besoins de chaque étude et chaque utilisateur.

Enfin, une conclusion termine ce travail.

En annexe, nous présentons les programmes développés à l'aide de logiciel MATLAB ceci de permis de réaliser les illustrations numériques fournis dans le quatrième chapitre.

CHAPITRE 1

GENERALITES SUR LA METHODOLOGIE DE LA RECHERCHE EXPERIMENTALE

Les plans d'expériences permettent d'organiser au mieux les essais qui accompagnent une recherche scientifique ou des études industrielles. Ils sont applicables à de nombreuses disciplines et à toutes les industries à partir du moment où l'on cherche le lien qui existe entre une grandeur d'intérêt « y » et des variables « x_i » [6]. Il faut penser aux plans d'expériences si l'on s'intéresse à une fonction du type :

$$y = f(x_i)$$

L'utilisation des plans d'expériences permet d'obtenir un maximum de renseignements avec un minimum d'expériences. Pour cela, il faut suivre des règles mathématiques et adopter une démarche rigoureuse. Il existe des nombreux plans d'expériences adaptés à tous les cas rencontrés par un expérimentateur. Nous aborderons dans ce chapitre, la modélisation d'une réponse par la méthode de planification des expériences. La compréhension de la méthode des plans d'expériences s'appuie sur deux notions essentielles, celle d'espace expérimental et celle de modélisation mathématique des grandeurs étudiée.

1.1 Notion d'espace expérimental

Un expérimentateur qui lance une étude s'intéresse à une grandeur qu'il mesure à chaque essai. Cette grandeur s'appelle la réponse, c'est la grandeur d'intérêt. La valeur de cette grandeur dépend de plusieurs variables. Au lieu du terme «variable» on utilisera le mot facteur. La réponse dépend donc de un ou de plusieurs facteurs. Lorsqu'on étudie l'influence d'un facteur, en général, on limite ses variations entre deux bornes. La borne inférieure est le niveau bas. La borne supérieure est le niveau haut [7].

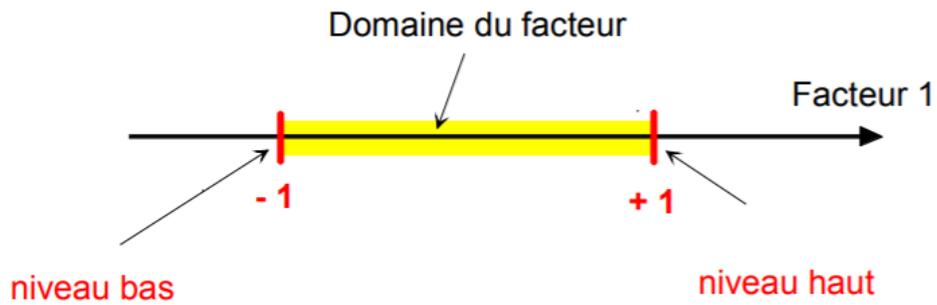


Figure 1.1 : Le niveau bas du facteur est noté par - 1 et le niveau haut par +1. Le domaine de variation du facteur est constitué de toutes les valeurs comprises entre le niveau bas et le niveau haut

On définit, comme pour le premier facteur, son niveau haut, son niveau bas et son domaine de variation. Ce second axe est disposé orthogonalement au premier. On obtient ainsi un repère cartésien qui définit un espace euclidien à deux dimensions. Cet espace est appelé l'espace expérimental [7].

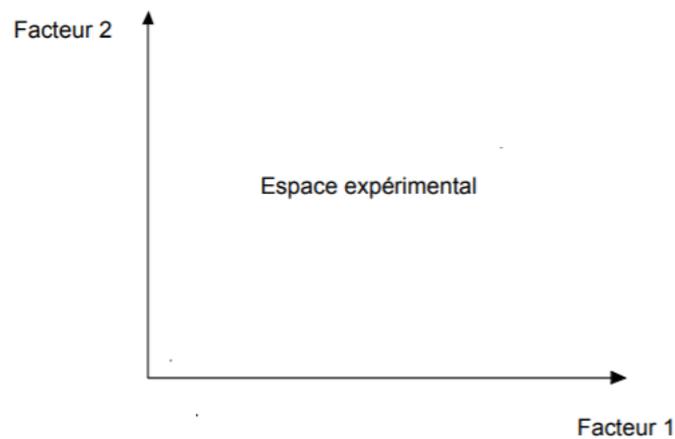


Figure 1.2 : Chaque facteur est représenté par un axe gradué et orienté. Les axes des facteurs sont orthogonaux entre eux. L'espace ainsi défini est l'espace expérimental.

Le regroupement des domaines des facteurs définit le «domaine d'étude». Ce domaine d'étude est la zone de l'espace expérimental choisie par l'expérimentateur pour faire ses essais. Une étude, c'est-à-dire plusieurs expériences bien définies, est représentée par des points répartis dans le domaine d'étude.

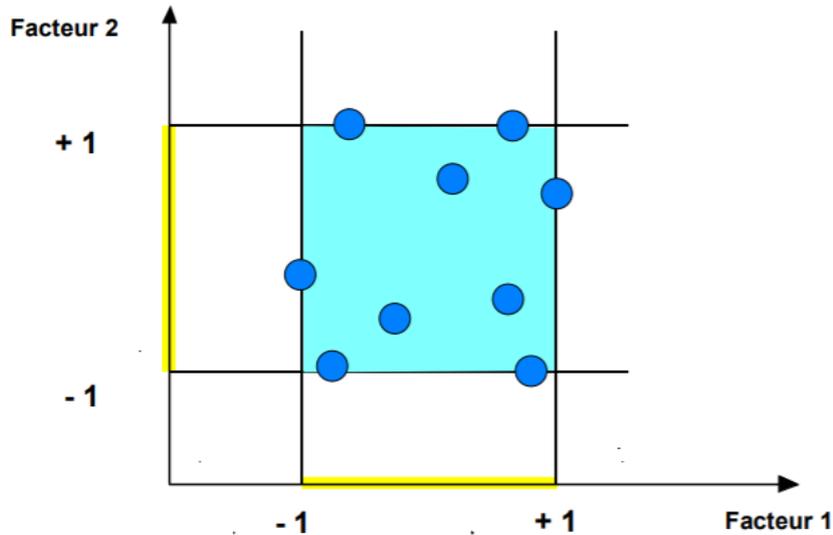


Figure 1.3 : Domaine d'étude de deux facteur

1.2 Notion de modélisation mathématique

On choisit a priori une fonction mathématique qui relie la réponse aux facteurs. On prend un développement limité de la série de Taylor-Mac Laurin. Les dérivées sont supposées constantes et le développement prend la forme d'un polynôme de degré plus ou moins élevé :

$$y = a_0 + \sum a_i x_i + \sum \sum a_{ij} x_i x_j + \dots + \sum a_{ii} x_i^2 + \sum a_{ij\dots k} x_i x_j \dots x_k \quad (1.1)$$

Où

- y est la réponse ou la grandeur d'intérêt.
- x_i représente le niveau attribué au facteur i par l'expérimentateur pour réaliser un essai.
- a_0, a_i, a_{ij}, a_{ii} sont les coefficients du modèle mathématique adopté a priori.

Si l'on mesure plusieurs fois une réponse en un même point expérimental, on n'obtient pas exactement le même résultat. Les résultats sont dispersés [8]. Les dispersions ainsi constatées sont appelées erreurs expérimentales, donc le modèle utilisé par l'expérimentateur s'écrit alors :

$$y = a_0 + \sum a_i x_i + \sum \sum a_{ij} x_i x_j + \dots + \sum a_{ii} x_i^2 + \sum a_{ij\dots k} x_i x_j \dots x_k + \varepsilon \quad (1.2)$$

Avec ' ε ' représente le manque d'ajustement et l'erreur expérimentale.

Chaque point expérimental permet d'obtenir une valeur de la réponse. Cette réponse est modélisée par un polynôme dont les coefficients sont les inconnues qu'il faut déterminer. A la fin du plan d'expériences, on a un système de n équations (s'il y a n essais) à p inconnues (s'il y a p coefficients dans le modèle choisi a priori). Ce système s'écrit d'une manière simple en notation matricielle

$$y = XA + \varepsilon \quad (1.3)$$

Où

- y est le vecteur des réponses.
- X est la matrice de calcul, ou matrice du modèle,
- A est le vecteur des coefficients.
- e est le vecteur des écarts.

Ce système possède un nombre d'équations inférieur au nombre d'inconnues. Il y a n équations et $(p + n)$ inconnues. Pour le résoudre, on utilise une méthode de régression basée sur le critère des moindres carrés. On obtient ainsi les estimations des coefficients que l'on note :

$$\hat{A} = ({}^tXX)^{-1} {}^tXY$$

Avec,

- tX est la matrice transposée de X ,
- $({}^tXX)$ est la matrice d'information,
- $({}^tXX)^{-1}$ est la matrice de dispersion.

1.3 Quelques notions sur les plans d'expériences usuels

Nous allons maintenant appliquer les notions et les propriétés que nous venons de décrire aux plans d'expériences les plus classiques. Parmi ces plans, les plus courants sont les plans factoriels, les plans de Box-Bahnken et composites centrés qui sont faciles à générer... etc.

1.3.1 Plan fractionnaire

Un plan fractionnaire est un plan dans lequel les chercheurs n'effectuent qu'un sous-ensemble déterminé ou une "fraction" des essais du plan factoriel complet. Les plans factoriels fractionnaires constituent un bon choix lorsque les ressources sont limitées ou lorsque le nombre de facteurs figurant dans le plan est important du fait du nombre moins

élevé d'essais effectués par rapport aux plans factoriels complets. Un plan factoriel fractionnaire utilise un sous-ensemble d'un plan factoriel complet, ainsi, certains des effets principaux et des interactions à 2 facteurs sont confondus et ne peuvent pas être séparés des effets des autres interactions plus élevées. En général, les chercheurs veulent supposer que les effets plus élevés sont négligeables afin d'obtenir des informations sur les effets principaux et les interactions faibles avec moins d'essais [9].

1.3.2 Plan factoriel complet

Un plan factoriel complet est un plan dans lequel les chercheurs mesurent les réponses pour toutes les combinaisons de niveaux de facteurs. Il y a deux types de plans factoriels complets :

- Des plans factoriels complets à 2 niveaux comprenant uniquement des facteurs à 2 niveaux.
- Des plans factoriels complets généraux comprenant des facteurs à plus de deux niveaux.

Le nombre d'essais nécessaires dans le cadre d'un plan factoriel complet à deux niveaux est égal à 2^k où k correspond au nombre de facteurs. Plus le nombre de facteurs figurant dans un plan factoriel à deux niveaux augmente, plus le nombre d'essais nécessaires pour effectuer un plan factoriel complet augmente rapidement. Par exemple, un plan factoriel complet à 2 niveaux avec 6 facteurs requiert 64 essais, un plan avec 9 facteurs requiert 512 essais. Un plan factoriel fractionnaire de fraction 1/2 ne requiert que la moitié de ces essais [9].

Exemple : plan factoriel complet à 2 niveaux à 2 facteurs

La vitesse de propagation des ultrasons dans un fluide dépend de la température et de la nature du fluide. L'étude consiste à étudier l'influence de la température et de la concentration en éthanol d'un mélange eau-alcool sur la vitesse de propagation. On effectue pour cela un plan factoriel complet à 2 facteurs ayant 2 niveaux, noté : 2^2 . Le domaine d'étude est un carré dont les sommets A, B, C et D sont les meilleurs emplacements.(voir Figure 1.4)

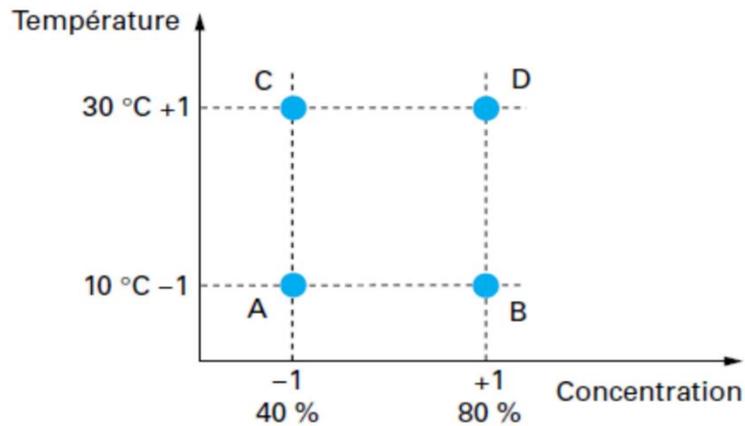


Figure 1.4 : Meilleurs emplacements des points expérimentaux

1.3.3 Plan de box-bahnken

Un plan de Box-Bahnken est un plan dont les variables ne prennent que 3 niveaux $(-a, 0, +a)$, si l'on considère pour le domaine expérimental l'hypercube $[-1,1]^p$. Ces plans d'expériences sont composés d'une matrice factorielle à 2 niveaux (2^p points) et de blocs incomplets équilibrés en suivant un arrangement particulier. Ces matrices sont ensuite complétées d'un point au centre du domaine. Ces plans sont une alternative aux plans composites parce qu'ils requièrent seulement 3 niveaux pour chaque facteur et permettent aussi de modéliser une surface de réponse de degré 2. On peut trouver dans la référence numéro 9, ou bien aussi dans la référence n° 10 la méthode de construction de ces plans et plus précisément la manière de former les blocs. En termes de nombre de points, un plan de Box-Bahnken est comparable à un plan composite en dimension 3 et 4. A noter qu'il n'existe pas de Box-Bahnken en dimension 2. Ces plans ayant leurs points sur les axes factoriels du domaine, ils ne remplissent pas correctement l'espace.

Remarque : Les plans de Box-Bahnken pour 3 facteurs impliquent trois blocs, dans chacun, un plan factoriel 2^2 qui est représenté par chaque paire de traitement pendant que le troisième facteur reste fixé à 0. Donc le plan de Box-Bahnken de 3 facteurs possède 15 essais (12 arêtes et 3 points au centre).

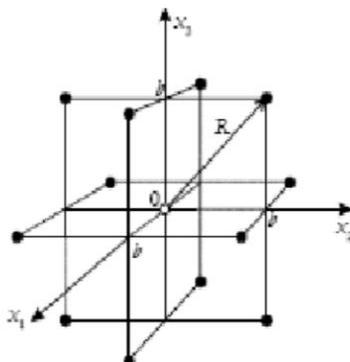


Figure 1. 5 : Plan de Box-Bahnken pour 3 facteurs

1.3.4 Plan composite

Une matrice d'expériences composite est une combinaison

- D'une matrice d'expériences factorielle à 2 niveaux, complète (2^p) ou fractionnaire (2^{p-k}), dont les points sont pris parmi les sommets de l'hypercube ($[-1,1]$ par exemple).
- D'une matrice d'expériences axiale composée de points disposés symétriquement sur chaque axe à une distance α du centre du domaine.
- Et d'un point central pour p facteurs qui permet d'obtenir une information sur la variabilité du phénomène ou tester la validité du modèle (par exemple, dans le cas d'un modèle linéaire de degré 1, tester la présence d'une courbure).

Un composite à faces centrées sur le domaine cubique $[-1,1]^2$ correspond donc à un plan factoriel à 3 niveaux $(-1, 0, 1)$. A noter que ces plans sont adaptés à la démarche « one-at-a-time » qui va imposer des points sur les axes et les plans factoriels 2^p [11].

1.4 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) ou (1.2) , les modèles déterministes sont caractérisés par l'absence d'un terme d'erreur [12].

1.4.1 Plan marginaux

Nous introduisons dans cette section des plans qui, par construction, ont des bonnes propriétés en termes de non redondance, et de non alignement sur certains sous-espaces. Cependant, rien ne garantit qu'ils remplissent bien l'espace [13].

On définit par ailleurs ce que l'on entend par marges. La notion de marges se réfère à des sous-espaces factoriels. Par exemple, les marges de dimension 1 correspondent aux axes factoriels.

1.4.1.1 Hypercube latin

Définition 1.1: Un hypercube latin à n points et p variables est une matrice $n \times p$ dont chacune des colonnes est une permutation de l'ensemble $\{1, 2, \dots, n\}$. On le note LHD (n, p).

La méthode d'échantillonnage par hypercube latin a été introduite pour évaluer numériquement les intégrales multiples. Elle permet d'assurer la non redondance de l'information à travers d'une bonne répartition des projections sur les axes factoriels. En pratique, les hypercubes latins sont très utilisés en planification d'expériences numériques notamment pour leur simplicité d'usage et de construction [13].

Ainsi, pour un plan LHS à n observations, chaque dimension de l'espace sera découpée en n intervalles et un point par intervalle sera choisi (et ce pour chaque dimension). Par exemple, sur \square^2 , un LHS à 5 points et un LHS à 10 points peuvent donner les plans suivants :

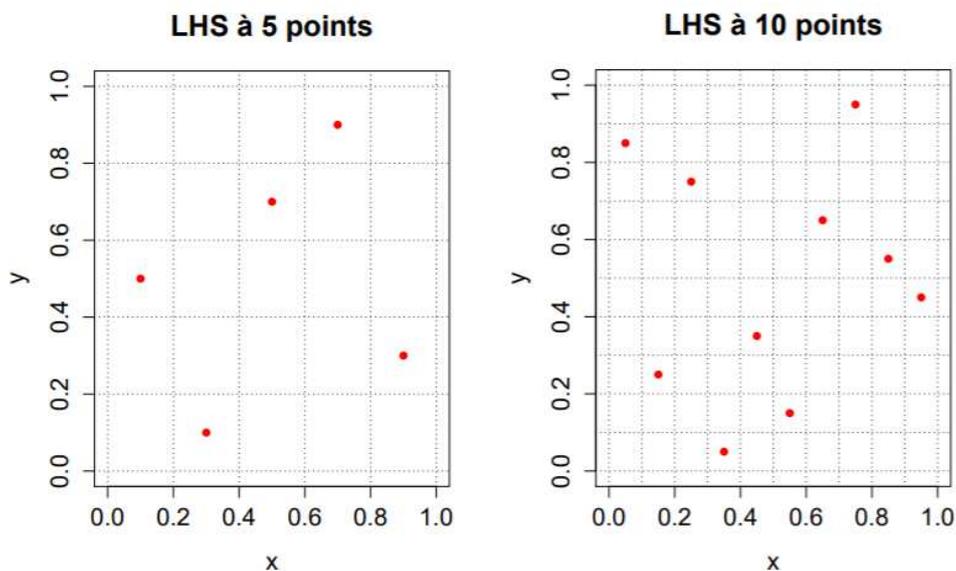


Figure 1.6: Plans LHS avec 5 et 10 points

Ces plans sont très faciles à générer et permettent une projection uniforme sur les marginales. Nous pouvons aussi rajouter du brouillage qui induit que chaque point est tiré selon la loi $U(0, 1)$ dans son intervalle afin de ne pas avoir que des points centres dans chaque intervalle.

Ces plans ne sont pas uniques car il existe de nombreuses combinaisons possibles pour les plans LHS “simples”. Ils peuvent aussi être de très mauvaise qualité, ce qui est le cas quand nous obtenons un plan LHS composé de tous les points d’une des diagonales [6].

1.4.1.2 Tableaux orthogonaux

Définition 1.2 : Un tableau orthogonal de force t à q symboles est une matrice de n lignes et $(k > t)$ colonnes avec ses éléments prenant q valeurs possibles, telle que toutes les sous matrices de taille $(n \times t)$ aient q^t lignes apparaissant le même nombre λ fois.

Conceptuellement, les tableaux orthogonaux sont très proches des hypercubes latins. En effet, ils ont l’avantage d’avoir les bonnes propriétés de projection en grande dimension qu’ont les hypercubes latins en dimension 1 [13].

Il est évident que $n = \lambda q^t$. On note un tel tableau orthogonal $OA_\lambda(n, k, q, t)$.

Géométriquement, cela revient à découper les axes du cube unité en q segments égaux afin d’obtenir q^d cellules de même taille. On choisit n cellules de telle façon qu’elles forment un tableau orthogonal défini comme ci-dessus. Cette définition assure que chaque t colonnes de la matrice du plan, (soit chaque t -uplet), apparaît exactement λ -fois.

Remarque : Un tableau orthogonal de force 1 est un hypercube latin.

Comme pour les hypercubes latins, on peut choisir le point aléatoirement dans la cellule mais aussi au centre de la cellule. Dans ce dernier cas, toutes les projections sur les sous-espaces de dimensions t conduisent alors à une grille régulière comme on peut le constater sur la Figure 1.7.

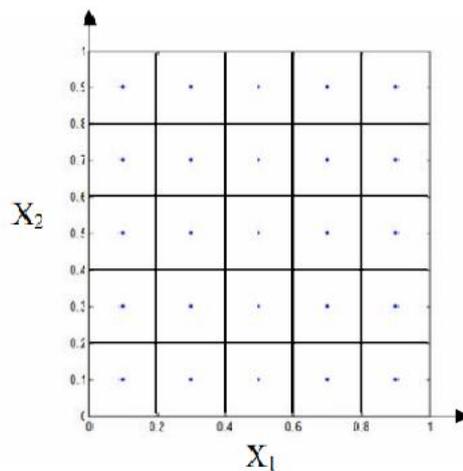


Figure 1.7 : Un plan généré par un tableau orthogonal $OA_1 (25, 5, 5, 2)$ dont les points sont centrés et projetés sur le sous-espace (X_1, X_2)

Définition 1.3 : Un échantillonnage par tableau orthogonal (abusivement appelé par la suite tableau orthogonal) à n points sur $[0,1]^d$ est un ensemble de points X^i tel que

$$X_j^i = \frac{\pi_j(A_j^{(i)}) + U_j^{(i)}}{q}; \quad 1 \leq i \leq n; \quad 1 \leq j \leq d$$

Où π_j est une permutation de $0, \dots, q-1$, $A_j^{(i)}$ sont les éléments du tableau orthogonal et $U_j^{(i)} \sim U[0, 1]$ est une valeur aléatoire de distribution uniforme sur $[0,1]$.

Ainsi, $(\pi_1(A_1^{(i)}), \dots, \pi_d(A_d^{(i)}))$ représente la cellule dans laquelle se trouve le point X^i , et $((U_1^i), \dots, (U_d^i))$ désigne à quel endroit le point se trouve dans la cellule.

Le tableau orthogonal correspond à la matrice de n lignes et d colonnes à coefficients X_j^i .

Si on effectue une permutation des symboles sur chaque colonne d'un tableau orthogonal de force t , alors on obtient toujours un tableau orthogonal de force t . La Figure 1.8 montre les points générés par un tableau orthogonal de type $OA_1 (25, 5, 5, 2)$ projetés sur le sous-espace. Nous pouvons observer qu'il y a bien un unique point par cellule et donc que les points projetés sur le sous-espace sont répartis uniformément.

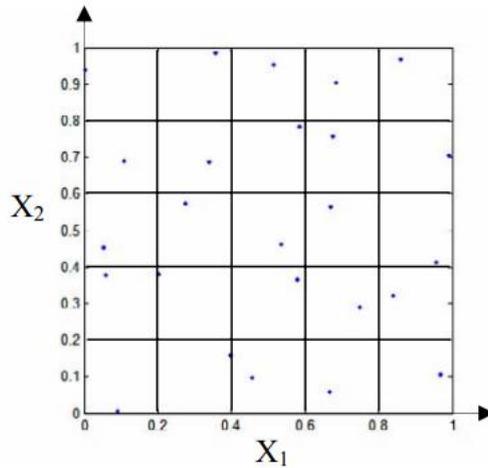


Figure 1.8 : Un plan généré par un tableau orthogonal $1 OA_1 (25, 5, 5, 2)$ dont les points sont randomisés et projetés sur le sous-espace (X_1, X_2)

1.4.1.3 Hypercubes latins basés sur les tableaux orthogonaux

Définition 1.4: 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 $B = \{B_i\}_{i=1}^n$ le tableau

résultant. Un hypercube latin basé sur un tableau orthogonal est l'ensemble de points $P = \{X_i\}_{i=1}^n$ défini par :

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

Où $U_{i,j} \ j \in D$ 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 un tableau orthogonal jouit d'une bonne stratification des projections d'ordre 1 et t. [13]

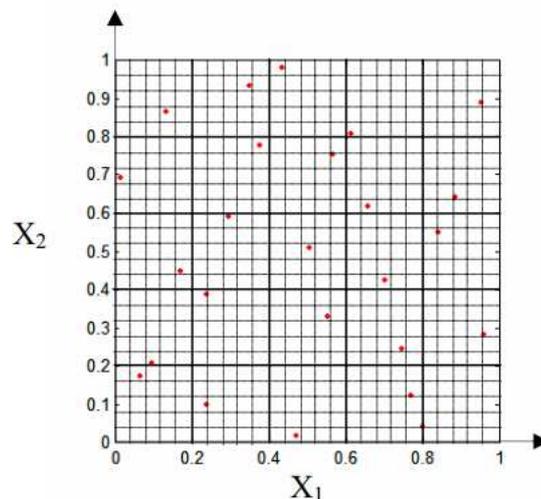


Figure 1.9 Un hypercube latin généré par un tableau orthogonal $OA_1(25, 5, 5, 2)$ dont les points sont randomisés et projetés sur le sous-espace (X_1, X_2)

1.4.2 Suites à faible discrédance

Nous présentons ici des plans dont l'objectif est de bien remplir l'espace. Aussi, nous nous attachons à regarder leurs propriétés en projection. Ces suites de points sont initialement utilisées pour remplacer les suites aléatoires dans la méthode de Monte-Carlo, d'où la dénomination de méthodes de quasi Monte-Carlo. La plupart de ces suites à faible discrédance [14, 15, 16, 17] sont générées par des algorithmes déterministes afin d'obtenir des points répartis le plus uniformément possible dans le domaine expérimental. Par définition, la discrédance mesure l'écart entre une distribution de points uniforme et une distribution de points donnée, en d'autres termes elle permet de mesurer l'irrégularité de la

distribution. Si (dans le cas de la dimension 1) l'on considère la fonction de répartition empirique \widehat{F}_n des points x^0, \dots, x^{n-1} . On peut définir la discrédance par [13] :

$$D_n(\mathbf{X}) = \sup_{x \in [0,1]} |\widehat{F}_n(x) - F_U(x)|$$

Avec F_U la fonction de répartition de la loi uniforme sur $[0,1]$.

$D_n(\mathbf{X})$ est la statistique de Kolmogorov-Smirnov permettant de tester l'adéquation à la loi uniforme.

1.4.2.1 Les suites de van der Corput

Définition 1.5 : Soit un entier $b \geq 2$. La suite $C_b = \{x^0, x^1, \dots\} \subset I$ donnée par $x^i = \varphi_b(i)$ est appelée suite de van der Corput en base b .

Les suites de van der Corput sont des suites à discrédance faible dans l'intervalle unité $I = [0, 1]$. Tout entier $b \geq 2$ (un tel nombre est appelé une base) peut être utilisé pour représenter n'importe quel $i \in \mathbb{N}$ de manière unique à l'aide d'une suite de coefficients $\square_b = \{0, 1, \dots, b - 1\}$:

$$i = \sum_{k=1}^{\infty} C_k(i) b^{k-1} \text{ avec } C_k(i) \in \square_b, \text{ pour tout } k \in \mathbb{N}^*$$

On remarque que les coefficients $C_k(i)$ sont nuls pour tout $k > 1 + \lfloor \log_b i \rfloor$ si bien que la somme ci-dessus est en fait finie. On se sert de cette représentation pour définir la fonction radicale inverse φ_b :

$$\varphi_b(i) = \sum_{k=1}^{\infty} C_k(i) b^{-k}, \text{ pour tout } i \in \mathbb{N}$$

On s'aperçoit facilement que $\varphi_b(i) \in [0, 1]$ pour tout $i \in \mathbb{N}$. [4]

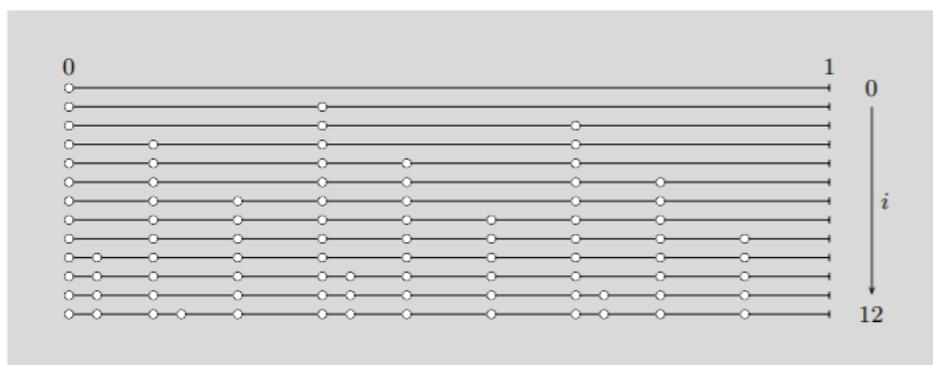


Figure 1.10 Les 13 premiers points de la suite de van der Corput en base 3.

1.4.2.2 Les suites de Halton

Définition 1.6: Soit b_1, \dots, b_s des entiers positifs premiers entre eux deux à deux. La suite $H_{b_1, \dots, b_s} = \{x^0, x^1, \dots\}$ donnée par $x^i = (\varphi_{b_1}(i), \dots, \varphi_{b_s}(i)) \in I$, est appelée suite de Halton en bases b_1, \dots, b_s .

Les suites de Halton (voir figure 1.11) sont des généralisations en dimension quelconque $s \geq 1$ des suites de van der Corput. L'idée consiste à considérer la fonction radicale inverse dans différentes bases simultanément. Halton a démontré que ses suites sont à discrédance faible.

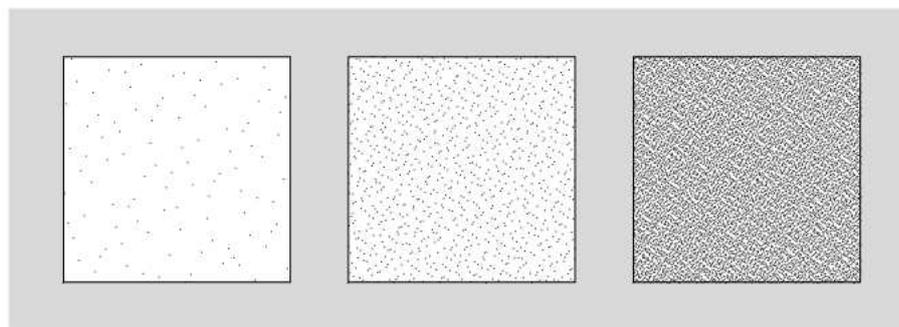


Figure 1.11 : Les 100, 1 000 et 10 000 premiers points d'une suite de Halton en bases 2 et 3.

Ainsi, pour tout $j \in \{1, \dots, s\}$, la suite constituée des composantes $\{x_j^0, x_j^1, \dots\}$ des points d'une suite de Halton $H_{b_1, \dots, b_s} = \{x^0, x^1, \dots\}$ n'est autre que la suite de van der Corput en base b_j .

Ces suites ont l'avantage d'être faciles à implémenter et d'avoir un temps d'exécution très faible. Pour passer de $x^i = \varphi_{b_1}(i)$ à $x^{i+1} = \varphi_{b_1}(i+1)$ il suffit d'effectuer une addition, en base b , à droite de $\frac{1}{b}$.

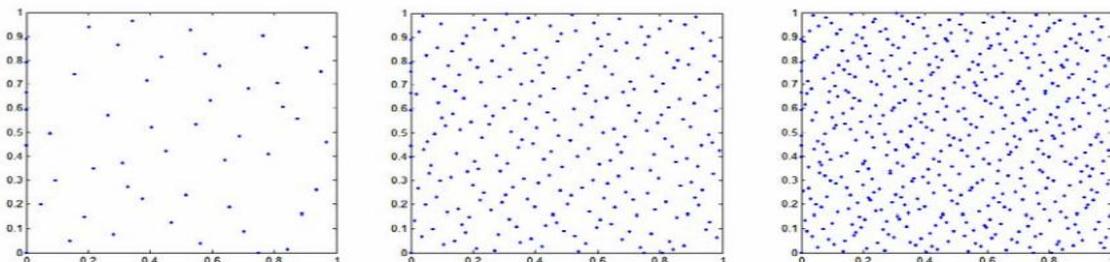


Figure 1.12 : (De la gauche vers la droite) Les 50, 250 et 500 premiers points d'une suite de Halton en bases 2 et 3

Malgré l'uniformité de la distribution dans le cube unité, ces suites présentent des pathologies en dimension élevée. En effet, lorsque l'on projette les points d'une telle suite dans le carré unité, l'allure de la structure est parfois décevante (Figure 1.13). Nous pouvons observer des diagonales décalées à chaque itération, qui laissent place à de nombreuses zones lacunaires dans le carré unité. Ce phénomène résulte de la régularité des sous-suites de longueur b des suites de Van der Corput. A noter cependant que cette pathologie ne se manifeste pas pour n'importe quelle paire de bases. En général, il est plus fréquent d'observer ce phénomène en dimension élevée et lorsque la différence entre les deux bases est très petite par rapport aux bases utilisées.

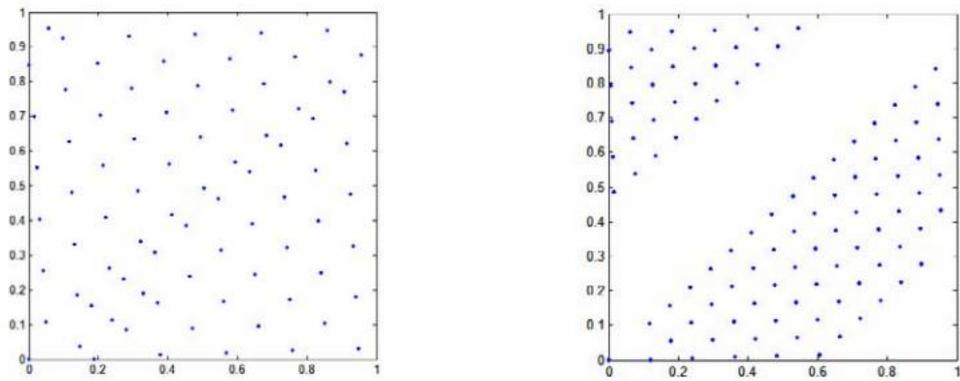


Figure 1.13 : Des mauvaises projections en 2D. A gauche : les 80 premiers points d'une suite de Halton en bases 11 et 13. A droite : les 100 premiers points d'une suite de Halton en bases 17 et 19

1.4.2.3 Les suites de Hammersley

Définition 1.7 : Soit b_1, \dots, b_{s-1} des entiers positifs premiers entre eux deux à deux et $n \in \mathbb{N}$. La séquence $H_{b_1, \dots, b_{s-1}}^n = \{x^0, \dots, x^{n-1}\}$ donnée par :

$$x^i = \left(\frac{i}{n}, \varphi_{b_1}(i), \dots, \varphi_{b_{s-1}}(i) \right) \in \mathbf{I}^s$$

Est appelée suite de Hammersley en bases b_1, \dots, b_{s-1} .

Une suite de Hammersley en dimension est constituée d'un ensemble fini de points directement issus d'une suite de Halton en dimension $s - 1$.

Ces suites présentent le même phénomène de diagonales successives. De plus, il est impossible de rajouter des points supplémentaires à ces suites sans pour autant perturber la discrédance. Dans le cas où le nombre de points à générer n'est pas connu à

l'avance, il est déconseillé d'utiliser une suite de Hammersley. Aussi, ces suites ont perdu l'aspect itératif des suites de Halton qui permettait de rajouter des points facilement.

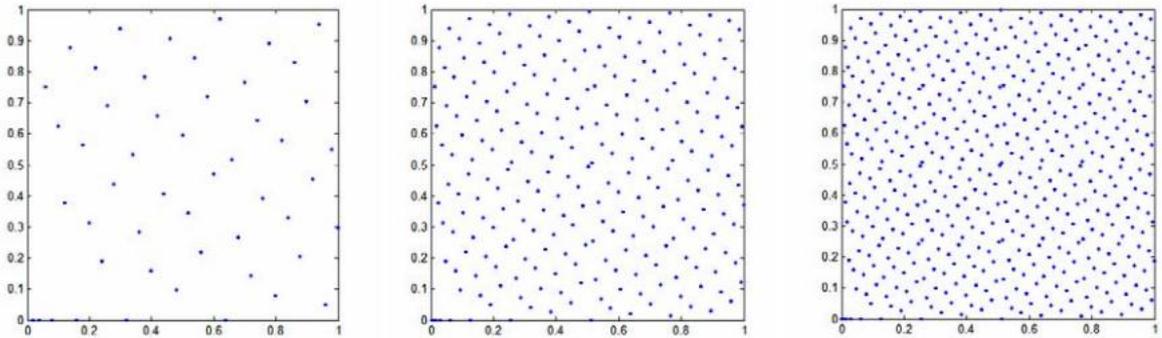


Figure 1.14 : (De la gauche vers la droite) Les 50, 250 et 500 premiers points d'une suite de Hammersley en base 2

1.4.2.4 Suites de Faure

Les suites de Faure sont définies à partir de la fonction radicale inverse, ϕ_b , est une matrice génératrice de Pascal $C = (c_{kl})$ donnée par

$$\binom{l-1}{k-1} = \begin{cases} \frac{(l-1)!}{(k-1)!(l-k)!} & \text{si } k \leq l \\ 0 & \text{sinon} \end{cases} \quad \forall l, k \in \mathbb{N}^*$$

Définition 1.8 Soit $b \geq d$ un nombre premier. La suite de Faure $F = \{x^0, \dots, x^{n-1}\}$ en dimension d est définie comme suit

$$x_j^i = \phi_b(C^{j-1}i)$$

Où $C^{j-1} = \binom{l-1}{k-1} (j-1) \bmod b$, est la matrice génératrice de la j -ième dimension d'une suite de Faure en dimension d .

Ces suites sont considérées comme meilleures que celles de Halton et de Sobol. La figure ci-dessous montre une distribution de 50, 250 et 500 points d'une suite de Faure pour deux dimensions.

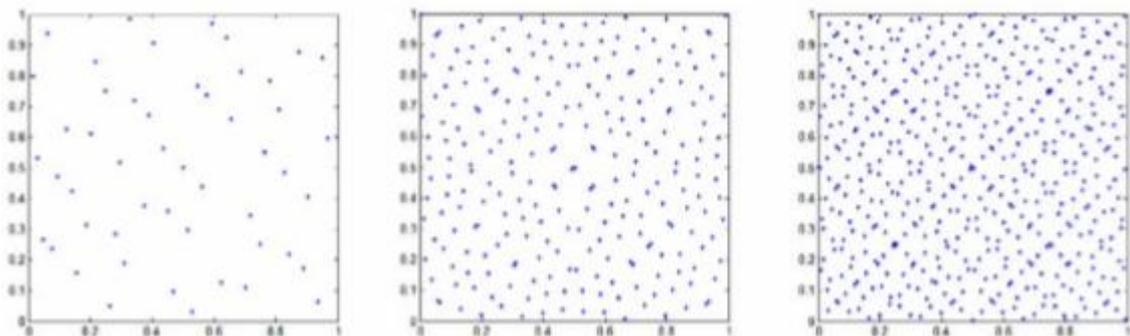


Figure 1.15 : les 50, 250 et 250 premiers points d'une suite de Faure.

1.5 Critères d'optimalités pour les plans d'expériences

L'évaluation de la qualité de la structure d'un ensemble de points issu d'une base de données ou d'un plan d'expériences nécessite l'utilisation de critères quantitatifs. Il existe de nombreux critères qui permettent, entre autres, d'évaluer la qualité d'un plan d'expériences. En général, pour les plans de surface de réponse nous recherchons le placement optimal des points d'expériences pour lequel l'erreur sur les réponses prédites est la plus faible possible (critère d'orthogonalité, critère de presque d'orthogonalité, critère D, critère A...etc) et pour le cas des plans de space filling nous cherchons l'uniformité d'une répartition de points (tels que le critère de discrédance, critère de distance, critère de recouvrement) [18].

- **D-optimalité** : Une matrice d'expériences est dite optimale-D, si elle conduit au déterminant minimal pour sa matrice de dispersion $(X^T X)^{-1}$.
- **G-optimalité** : Le critère-G prend en compte la plus grande valeur, sur tout le domaine expérimental, de la fonction de variance engendré par la matrice X . La meilleure matrice d'expériences vis à vis de ce critère étant celle qui a la plus petite valeur.
- **Critère d'orthogonalité** : Une matrice d'expériences est orthogonale si elle permet d'obtenir des estimations des coefficients indépendants. Cela se caractérise par des axes de l'ellipsoïde parallèles aux axes des coefficients. Cette propriété est obtenue quand $(X^T X)^{-1}$ est diagonale donc quand les covariances des coefficients sont nulles.
- **Critère de Presque Orthogonalité** : Si la matrice $(X'X)^{-1}$ obtenue en retirant sa première ligne et sa première colonne est diagonale, le critère de presque orthogonalité est respecté.
- **Critère d'isovariance par rotation** : On désire que les réponses calculées avec le modèle issu du plan d'expériences aient une erreur de prévision identique pour des points situés à la même distance du centre du domaine d'étude. Dans ce cas on parle de plan isovariant par rotation (rotable).
- **Critère de distance** (Mindist) [19] : représente la plus petite distance entre une paire parmi un plan de n points, ce critère est défini par

$$Mindist = \min_{1 \leq i \leq n} \min_{j \neq i} d(x_i, x_j)$$

Où, $d(x_i, x_j)$ est la distance Euclidienne entre le point x_i et x_j .

Une valeur plus élevée de Mindist devrait correspondre à une dispersion plus régulière des points du plan.

- **Critère de recouvrement** (cov) : permet de mesurer l'écart entre les points du plan et ceux d'une grille régulière. Ce critère est nul pour une grille régulière. L'objectif est donc de le minimiser pour se rapprocher d'une grille régulière, et ainsi assurer le remplissage de l'espace, sans toutefois l'atteindre pour respecter une distribution uniforme notamment en projection sur les axes factoriels [20] :

$$cov = \frac{1}{\bar{\delta}} \sqrt{\frac{1}{n} \sum_{i=1}^n (\delta_i - \bar{\delta})^2}$$

Où $\delta_i = \min_{i \neq j} (dis(x_i, x_j))$ et $\bar{\delta} = \frac{1}{n} \sum_{i=1}^n \delta_i$. Pour une grille régulière, $\delta_1 = \delta_2 = \dots = \delta_n$, alors $cov = 0$.

Dans le même contexte, on peut utiliser le ratio R , définie par :

$$R = \frac{\max_{1..n} \delta_i}{\min_{1..n} \delta_i}$$

Pour une grille régulière $R = 1$, Ainsi, plus R est proche de 1, et plus les points sont proches de ceux d'une grille régulière.

- **Critère de discrédance** (Disc) : La discrédance permet de mesurer l'écart entre la fonction de répartition empirique des points du plan et celle de la loi uniforme. Contrairement aux deux critères précédents, la discrédance n'est pas basée sur la distance entre les points. Il existe différentes mesures de discrédance. Nous retenons la discrédance en norme L2 [21].

$$Disc = \left(\frac{1}{3}\right)^p - \frac{2^{1-p}}{n} \sum_{i=1}^n \prod_{j=1}^p (1 - (x_j^i)^2) + \frac{1}{n^2} \sum_{i=1}^n \sum_{k=1}^n \prod_{j=1}^p (1 - \max(x_i^j, x_k^j))$$

CHAPITRE 2

PROCESSUS STOCHASTIQUES PONCTUELS

En probabilité et statistique, un processus ponctuel est un type particulier de processus stochastique pour lequel une réalisation est un ensemble de points isolés du temps et/ou de l'espace. Par exemple, la position des arbres dans une forêt peut être modélisée comme la réalisation d'un processus ponctuel.

Les processus ponctuels sont des objets très étudiés en probabilité et en statistique pour représenter et analyser des données spatialisées qui interviennent dans une multitude de domaines tels que l'écologie, l'astronomie, l'épidémiologie, la géographie, la sismologie, les télécommunications, la science des matériaux et beaucoup d'autres domaines.

Dans ce chapitre, nous rappelons dans un premier temps, quelques notions générales sur les processus ponctuels. Nous introduisons ensuite, quelques exemples de processus ponctuels de référence. Dans un deuxième temps, nous présentons les processus ponctuel marqués puis les processus ponctuels de Markov.

2.1 Quelques définitions et notation

On s'intéresse à des ensembles non ordonnés de points d'un espace donné χ , Par exemple \mathbb{R}^k .

Définition 2.1: On appelle configuration x un ensemble dénombrable, non ordonné de points de χ , tel que : $x = \{x_1, \dots, \dots x_n, \dots\}$ (voir Figure 2.1).

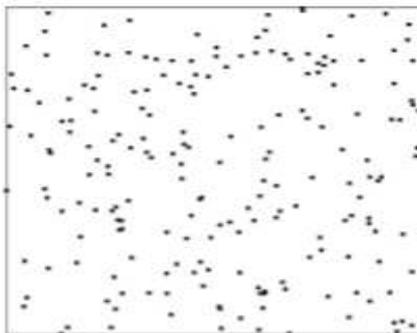


Figure 2.1 : Ensemble de points, ou configuration, de $\chi = [0,1]^2$.

Pour pouvoir considérer des configurations aléatoires, il faut ajouter des contraintes sur χ . On choisit de considérer un espace χ muni d'une métrique d tel que (χ, d) soit complet et séparable. Cette métrique permet de définir une topologie et une σ -algèbre borélienne. En pratique, χ sera souvent un sous-ensemble compact de \mathbb{R}^k muni de la distance euclidienne d .

Définition 2.2: On dit qu'une configuration $x \subseteq \chi$ est localement finie si dans tout borélien borné $A \subseteq \chi$ elle place un nombre $N_x(A)$ fini de points. La famille de toutes les configurations localement finies sera notée N^{lf} .

On définit alors la notion de processus ponctuel :

Définition 2.3: Un processus ponctuel sur χ est une application X d'un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ dans N^{lf} , telle que pour tout borélien $A \subseteq \chi$, $N(A) = N_x(A)$ est une variable aléatoire (voir Figure 2.2).

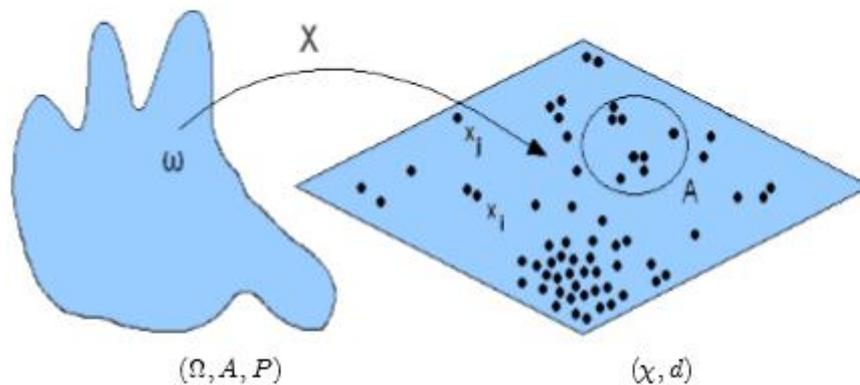


Figure 2.2 : Définition d'un processus ponctuel

Définition 2.4 : Si l'espace χ est borné ou si $N_x(\chi)$ est fini presque sûrement, le processus ponctuel est dit processus ponctuel localement fini.

Les réalisations d'un processus ponctuel X sont donc des configurations aléatoires de points tels que pour tout borélien $A \subseteq \chi$ le nombre de points dans A soit une variable aléatoire. Cela signifie qu'un processus ponctuel est une variable aléatoire à valeur dans l'espace mesurable (N^{lf}, N^{lf}) , où N^{lf} est la plus petite σ -algèbre telle que pour tout borélien borné $A \subseteq \chi$ l'application $x \rightarrow N_x(A)$ soit mesurable.

La mesure de probabilité induite sur N^{lf} est appelée la loi de X .

2.2 Loi d'un processus ponctuel

On appelle loi du processus ponctuel la mesure de probabilité π induite sur N^{lf} .

La loi d'un processus ponctuel X devrait être la mesure image par l'application X de P sur N^{lf} . Mais comme N^{lf} est définie par la mesurabilité des applications $x \rightarrow N_X(A)$ pour des boréliens $A \subseteq \chi$, l'analogie de la loi de probabilité pour des variables aléatoires dans un contexte de processus ponctuels est l'ensemble de lois jointes des vecteurs $(N(A_1), \dots, N(A_m))$ où les A_i sont des boréliens bornés.

Définition 2.5: La famille des lois marginales en dimensions finie (fidis) d'un processus Ponctuel X sur un espace métrique (χ, d) complet et séparable est la collection des lois jointes de $(N(A_1), \dots, N(A_m))$ pour tout vecteur fini (A_1, \dots, A_m) de boréliens bornés $A_i \subseteq \chi$, $i=1, \dots, m$ de longueur quelconque $m \in \mathbb{N}$.

Théorème 2.1: La loi d'un processus ponctuel X sur un espace métrique complet et séparable (χ, d) est entièrement déterminée par ses fidis. Donc, deux processus ponctuels partageant les mêmes fidis ont la même loi [22].

2.2.1 Processus ponctuel simple

On s'intéresse maintenant à la multiplicité éventuelle des points d'une configuration. En pratique, il est rare de travailler avec des processus ponctuels qui contiennent des points situés exactement au même endroit. Soit N_s^{lf} l'ensemble des configurations localement finies ne contenant que des points distincts. Pour vérifier que N_s^{lf} est N^{lf} -mesurable, il suffit de noter que, comme χ est séparable, il peut être recouvert par une union dénombrable de boules ouvertes $B(x_i, (\frac{1}{2})^i)$ de rayon arbitrairement petit. Par conséquent [22]:

$$\bigcup_{i \geq 0} \{ \omega \in \Omega : N(B(x_i, 2^{-j})) \in \{0,1\} \} \in N^{lf}$$

Définition 2.6 : Un processus ponctuel X est dit simple s'il prend ses valeurs dans N_s^{lf} presque sûrement.

Les processus ponctuels simples ont un avantage pratique. Ils ne nécessitent pas de connaître une famille entière de fidis pour disposer de leur loi. En effet, il suffit seulement de connaître les probabilités de vide $v(A) = P(N(A) = 0)$ pour une collection suffisamment grande de borélien A de χ [23].

Théorème 2.2 : La loi d'un processus ponctuel simple X sur un espace métrique complet et séparable (χ, d) est uniquement déterminée par les probabilités vides des boréliens bornés $A \subseteq \chi$.

2.2.2 Processus ponctuels finis

Cette finitude provient généralement de l'aspect borné de la fenêtre d'observation mais peut aussi provenir du fait que le processus génère simplement un nombre fini de points.

En plus d'être souvent simples, la plupart des configurations sont aussi finies.

2.2.3 Construction d'un processus ponctuel

Pour construire un processus ponctuel fini, on peut utiliser :

- a. Une distribution de probabilité discrète $(p_n)_{n \in \mathbb{N}}$ qui régit le nombre de points.
- b. Une famille de densités de probabilité symétriques $j_n(x_1, \dots, x_n)$, $n \in \mathbb{N}$, sur χ^n pour les positions des points.

Remarque

Le deuxième point implique que l'on suppose que χ soit équipé d'une mesure borélienne $v(\cdot)$ de manière à pouvoir définir la densité j_n par rapport à la mesure produit $v(\cdot)^n$.

2.3 Processus ponctuels marqués

Pour décrire des objets au lieu de points, il est simplement nécessaire d'ajouter à chaque point une marque qui décrit une ou des propriétés de l'objet.

Définition 2.7 : Soit (χ, d) et (K, d') deux espaces métriques, complets et séparables. Un processus ponctuel marqué dont les positions sont dans χ et les marques dans K est un

processus ponctuel sur $\chi \times K$, tel que le processus des points non marqués soit un processus ponctuel bien défini.

Pour comprendre cette définition, il faut d'abord voir que l'espace $\chi \times K$ est métrique, complet et séparable dès que l'on introduit la métrique :

$$\rho((x, k), (y, l)) = \max\{d(x, y), d'(k, l)\}$$

Dans les cas où l'espace des marques K est fini, on peut prendre, par exemple :

$$K = \{1, \dots, M\}, \quad d'(k, l) = |k - l|$$

Alors, tout processus ponctuel Y sur l'espace produit $\chi \times K$ fait apparaître un processus ponctuel de points non marqués X bien défini. Pour le comprendre, il nous faut considérer un borélien $A \subseteq \chi$. Alors, le nombre de points non marqués dans A peut-être écrit :

$$N_{X(A)} = \sum_{i=1}^{i=M} N_Y(A \times \{i\})$$

Par définition, chaque $N_Y(A \times \{i\})$ est une variable aléatoire finie, si bien que $N_Y(A)$ est aussi une variable aléatoire finie. Par contre, dans le cas où K est Continu, il faut faire plus attention. Par exemple, un processus de poisson sur \mathbb{R}^3 n'est pas un processus ponctuel marqué sur \mathbb{R}^2 avec ses marques dans \mathbb{R} , puisque $N_X(A) = N_Y(A \times \mathbb{R})$ n'est pas nécessairement fini pour un borélien $A \subseteq \mathbb{R}^2$ quelconque [24].

Exemple :

Sur la Figure 2.3, nous pouvons voir un exemple de processus marqué, processus ponctuel selon les marques $K = \mathbb{R}$, représenté par l'intermédiaire des différents plans, et processus ponctuel selon les points spatiaux de $\chi = \mathbb{R}^2$, représenté par les points sur chaque plan associé à une marque. Le processus non marqué résultant est obtenu par la projection des points spatiaux de chaque marque sur un plan de référence (plan défini par $\chi \times \{0\}$). Afin de ne pas alourdir le schéma, les projections des différents points marqués ne sont pas toutes explicitées.

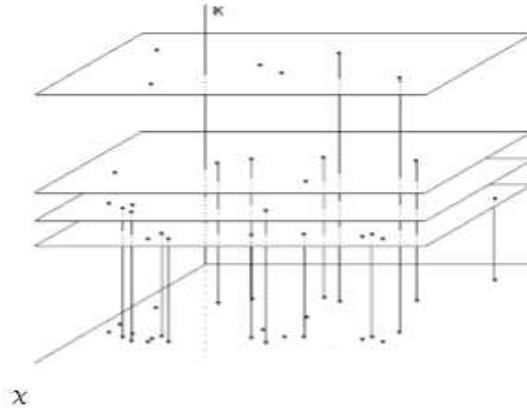


Figure 2.3 : Exemple d'un processus ponctuel marqué

2.4 Processus ponctuel de référence

Dans cette partie, χ désigne un compact de \mathbb{R}^k de volume $\mu(\chi) > 0$.

2.4.1 Processus ponctuel binomial

Soit χ un compact de \mathbb{R}^k de volume strictement positif $\mu(\chi)$. Un processus ponctuel binomial est défini comme l'union $X = \{X_1, \dots, X_n\}$ d'un nombre n fixé de points indépendants et uniformément distribués X_1, \dots, X_n . Comme $P(X_i = X_j) = 0$ pour tout $i \neq j$ alors le processus est simple. De plus, comme $P(N(\chi) = n) = 1$, le processus binomial est fini avec :

$$p_m = \begin{cases} 0 & \text{si } m \neq n \\ 1 & \text{si } m = n \end{cases}$$

Les points X_i sont uniformément distribués, donc :

$$j_n(X_1, \dots, X_n) = \left(\frac{1}{\mu(\chi)} \right)^n$$

Les j_n sont invariantes par permutation. Le processus binomial doit son nom au fait que pour tout borélien $A \subseteq \chi$, $N(A) = \sum_{i=1}^n 1_A(X_i)$ suit une loi binomiale de paramètres n et $\mu(A)/\mu(\chi)$.

2.4.2 Processus ponctuel de poisson (PPP)

Les processus ponctuels de Poisson sont les processus les plus élémentaires puisque, physiquement, ils traduisent la notion d'uniformité.

Définition 2.8: Soit $\nu(\cdot)$ une mesure borélienne sur un espace métrique et séparable (χ, d) tel que $\nu(\chi) > 0$ et $\nu(A)$ localement finie pour tout borélien borné A . Un processus ponctuel X sur χ est appelé un processus ponctuel de Poisson de mesure d'intensité $\nu(\cdot)$ si :

- **P1.** $N(A)$ suit une loi de Poisson d'espérance $\nu(A)$ pour tout borélien borné $A \subseteq \chi$.
- **P2.** Pour m boréliens disjoints A_1, \dots, A_m , les variables aléatoires $N(A_1), \dots, N(A_m)$ sont indépendantes. Si $\nu(\cdot)$ est non atomique, le processus de Poisson est simple.

Lorsque $\chi = \mathbb{R}^k$, on parle de processus de Poisson homogène lorsque la mesure d'intensité est $\lambda\mu(\cdot)$ où μ est la mesure de Lebesgue et λ un paramètre strictement positif. On appelle alors ce paramètre l'intensité du processus.

La propriété P2 peut être interprétée comme une propriété de non corrélation spatiale totale, puisqu'elle affirme que ce qui se passe à l'intérieur d'une fenêtre quelconque est totalement indépendant de ce qui se passe à l'extérieur. Le Théorème suivant précise cette idée, en affirmant que les points d'un processus Ponctuel de Poisson se comportent aléatoirement et n'interagissent pas les uns avec les autres [23].

Théorème 2.3 : Soit X un processus de Poisson ponctuel sur un espace métrique séparable complet (χ, d) de mesure d'intensité $\nu(\cdot)$ et A un borélien bornée sur χ . Alors, conditionnellement à l'événement $\{N(A) = n\}$, X restreint à A suit une loi binomial dont les points sont indépendants et disposés de manière ν -uniforme.

Nous pouvons justifier le résultat de ce dernier théorème, en considérant un borélien B inclus dans A . La probabilité de vide de B conditionnellement à l'événement $\{N(A) = n\}$ s'écrit :

$$\begin{aligned} \nu(B \setminus N(A) = n) &= P(N(B) = 0 \setminus N(A) = n) = \frac{P(N(B) = 0, N(A) = n)}{P(N(A) = n)} \\ &= \frac{P(N(B) = 0, N(A) - N(B) = n - 0)}{P(N(A) = n)} \\ &= \frac{P(N(B) = 0, N(A \setminus B) = n)}{P(N(A) = n)} \end{aligned}$$

X étant un processus ponctuel de Poisson, $N(B)$ et $N(A \setminus B)$ sont indépendants et suivent la loi de Poisson. Nous obtenons donc :

$$P(N(A \setminus B) = n) = \left(\frac{v(A \setminus B)}{v(A)} \right)^n$$

Qui coïncide avec la probabilité de vide $v(B)$ de n processus ponctuels indépendants et v -uniformes. B étant choisi de manière arbitraire, en utilisant le Théorème 2.2, on déduit le résultat.

Le processus ponctuel de Poisson joue un rôle fondamental dans l'étude des processus ponctuels. En effet, il sert de modèle de référence, mathématiquement étudiable, pour lequel il n'existe aucune interaction entre les points. De nombreux modèles, comme les processus ponctuels de Gibbs, par exemple, sont alors construits à partir du processus ponctuel de Poisson. Dans cette section, nous nous intéressons au processus de Poisson spécifié par une fonction d'intensité. Soit $\rho(\cdot)$ une fonction non négative localement intégrable sur $S \subseteq \mathbb{R}^k$, un processus ponctuel de Poisson d'intensité $\rho(\cdot)$ (que l'on note $PPP(\rho)$) est défini par [25] :

- Pour tout $A \in \mathcal{S}$ de mesure $0 < \int_A \rho(\xi) d\xi$ et finie, $N(A)$ suit une loi de Poisson de paramètre $\int_A \rho(\xi) d\xi$.
- Conditionnellement à $N(S)$, les points de $x \cap A$ sont distribués de façon *i.i.d* avec une densité proportionnelle à $\rho(\xi)$, $\xi \in A$:

$$P(N(A) = n) = \exp\left(-\int \rho(\xi) d\xi\right) \frac{\left(\int_A \rho(\xi) d\xi\right)^n}{n!}$$

La fonction $\rho(\cdot)$ caractérise entièrement la distribution de X .

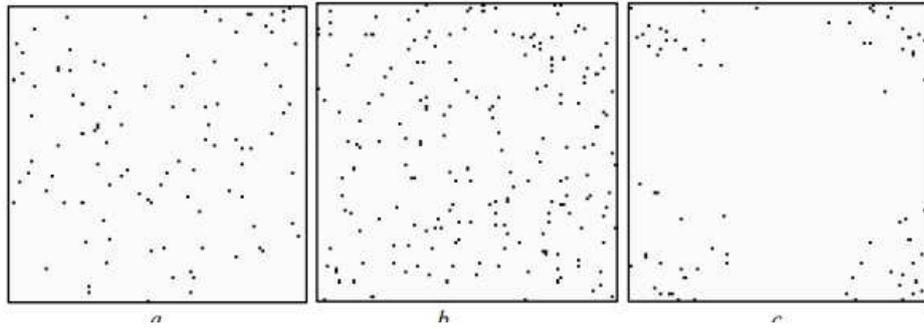


Figure 2.4 : Processus poisson sur une image 100×100 : a. d'intensité $\lambda=0,01$ b. d'intensité $\lambda=0,02$ c. processus non-homogène

2.4.3 Processus de Cox

Soit $Z = \{Z(\xi) : \xi \in S\}$ un champ aléatoire non négatif sur S localement intégrable presque-sûrement, pour tout borélien B borné, $\int_B Z(\xi)d\xi < \infty$. Si la distribution conditionnelle de X sachant Z est un processus de Poisson sur S avec fonction d'intensité Z , alors X est appelé un processus Cox gouverné par Z . La fonction d'intensité est définie par :

$$\rho(\xi) = EZ(\xi)$$

Et la fonction de corrélation de pair est définie par :

$$g(\xi, \eta) = \frac{E[Z(\xi)Z(\eta)]}{\rho(\xi)\rho(\eta)}$$

Par ailleurs les probabilités de vide sont définies par :

$$v(B) = E \exp \left(- \int_B Z(\xi) d\xi \right)$$

Pour tout borné $B \subseteq S$.

2.4.3.1 Processus ponctuel de Neumann-Scott

Pour le processus de Neyman-Scott, soit C un processus de Poisson stationnaire dans \mathbb{R}^k avec d'intensité $\kappa > 0$. Conditionnellement à C , soit $X_c, c \in C$, un processus de Poisson indépendant dans \mathbb{R}^k , où X_c admet une fonction d'intensité :

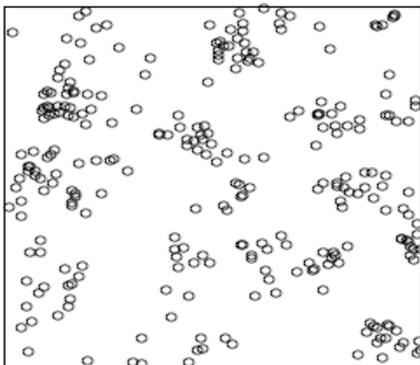
$$\rho_c(\xi) = \alpha q(\xi - c)$$

Où $\alpha > 0$ est un paramètre et q est un noyau (Pour tout $c \in \mathbb{R}^k, \xi \rightarrow q(\xi - c)$ est une fonction de densité). Alors le processus X est un processus de Neyman-Scott. Les points $c \in C$ sont appelés parents et ne sont pas observés. Les points de X sont appelés aussi les enfants du parent $c \in C$. Le processus X est en effet un processus de Cox dans \mathbb{R}^k gouverné par :

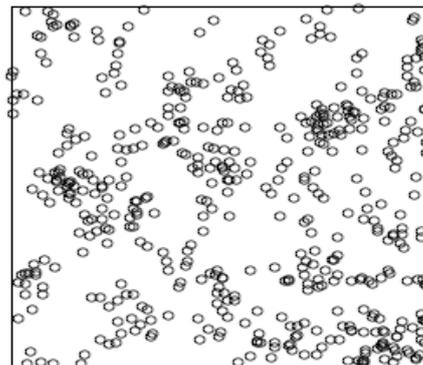
$$Z(\xi) = \sum_{c \in C} \alpha k(\xi - c)$$

Lorsque q est le noyau gaussien, X est appelé le processus de Thomas. La Figure 2.5 propose quelques réalisations de ce processus pour différentes valeurs des paramètres (l'écart-type des noyaux gaussiens) et κ [25].

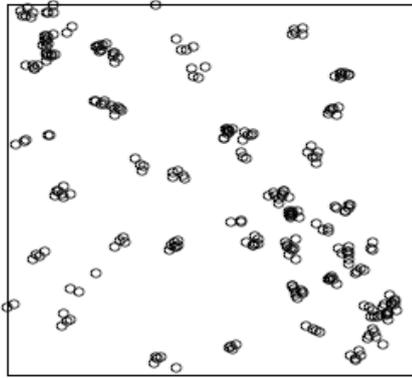
k = 50, s = 0.03, a = 5



k = 100, s = 0.03, a = 5



$k = 50, s = 0.01, a = 5$



$k = 100, s = 0.01, a = 5$

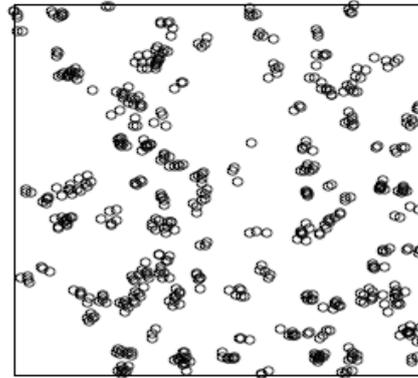


Figure 2.5 : Quatre réalisations d'un processus ponctuel de Thomas sur $[0, 1]^2$ pour différents paramètres

2.5 Processus ponctuel de Gibbs

On supposera dans ce paragraphe que S est une partie bornée de \mathbb{R}^k . Une façon d'un processus ponctuel est de définir sa densité de probabilité f relativement à celle d'un PP de Poisson homogène de densité 1. Cette approche permet en particulier de définir les PP de Gibbs. Généralement, la densité f d'un PP, n'est connue qu'à un facteur multiplicatif près, $f(x) = cg(x)$, où g est analytiquement connue. Ceci est sans conséquence quant à la simulation d'un PP par une méthode MCMC, méthode qui requiert la seule connaissance de g . Par contre, pour l'estimation par maximum de vraisemblance du modèle $f_{\theta}(x) = c(\theta)g_{\theta}(x)$, $c(\theta)$, qui est une constante analytiquement incalculable, devra être approchée par la méthode MCMC. Les processus ponctuels de Gibbs viennent de la physique statistique et caractérisent l'état d'équilibre des particules à l'intérieur de systèmes physiques fermés. Sa densité est donnée par :

$$f(x) = \frac{\exp(-V(x))}{Z}$$

où Z est une fonction de partition normalisant, en règle générale incalculable. V une fonction d'énergie, sera souvent dérivée d'un potentiel d'interaction γ :

$$V(x) = \sum_{\psi \subset x} \gamma(\psi).$$

2.6 Processus ponctuel de Markov

Nous présentons ici rapidement quelques définitions et propriétés des processus ponctuels de Markov. On considère une relation symétrique et réflexive \sim sur χ . On dit que deux points u et v de χ sont voisins si $u \sim v$. Par exemple, on peut considérer la relation de proximité sur $\chi = \mathbb{R}^d$: $u \sim v \Leftrightarrow d(u, v) \leq r$ pour un réel r quelconque.

Définition 2.9 : Le voisinage $\partial(A)$ d'un ensemble $A \subseteq \chi$ est défini comme l'ensemble : $\partial(A) = \{x \in \chi : \exists a \in A \quad x \sim a\}$.

Ripley et Kelly donnent la définition suivante d'un processus ponctuel de Markov [26].

Définition 2.10 : Soit (χ, d) un espace métrique complet et séparable, $\nu(\cdot)$ une mesure borélienne finie non atomique, et $\pi_\nu(\cdot)$ la loi d'un processus ponctuel de Poisson de mesure d'intensité $\nu(\cdot)$. Soit X un processus ponctuel sur χ défini par sa densité $p(\cdot)$ par rapport à $\pi_\nu(\cdot)$. Alors X est un processus ponctuel de Markov sous la relation symétrique et réflexive \sim sur χ si pour tout $x \in N^f$ telle que $p(x) > 0$

- a. $p(y) > 0$ pour tout $y \subseteq x$;
- b. pour tout $u \in \chi$, $p(x \cup \{u\})/p(x)$ ne dépend que de u et $\partial(\{u\}) \cap x = \{x \in x : u \sim x\}$.

La condition (a) traduit que si une configuration peut se produire, alors toutes les sous configurations qu'elle contient peuvent se produire aussi. Le quotient $\lambda(u, x)$ dans la condition (b) appelé l'intensité conditionnelle de Papangelou, est la densité de probabilité qu'il ait un point u sachant que x est réalisé ailleurs. Cette condition exprime une propriété de Markov locale : le comportement d'un point u par rapport à la configuration entière ne dépend que des ses proches voisins dans cette configuration.

Nous appelons une clique, une configuration dont tous les points sont voisins les uns des autres par rapport à une relation symétrique et réflexive notée \sim . L'ordre d'une clique est le nombre d'objets qu'elle contient. Le théorème suivant, équivalent de celui d'Hammersley-Clifford [23] pour les processus ponctuels, permet d'exprimer la densité d'un processus de Markov sous une forme énergétique décomposée sur les cliques de la configuration x :

Théorème 2.4 (Hammersley - Cliord). Une densité de processus ponctuel $f : E^f \rightarrow [0, \infty[$ est markovienne sous une relation de voisinage \sim si et seulement si il existe une fonction mesurable $\varphi : N^f \rightarrow [0, +\infty[$ telle que :

$$f(x) = \alpha \prod_{\text{cliques } y \subseteq x} \phi(y)$$

pour tout $x \in E^f$. $\phi(y)$ sont les potentiels d'interaction. Ce résultat permet de construire facilement des modèles markoviens, par exemple à interaction de paires :

2.7 Quelques propriétés générales des processus ponctuel

2.7.1 Mesure et théorème de Campbell

Il s'agit ici de définir des outils qui expriment des grandeurs moyennes ou d'ordre supérieur sur les processus ponctuels. Pour cela, le meilleur moyen est d'utiliser le fait que pour tout borélien borné A , $N(A)$ est une variable aléatoire, les moments de $N(A)$ fournissent des statistiques très utiles.

Définition 2.11 : Soit X un processus ponctuel sur un espace métrique complet et séparable (χ, d) . Pour tout borélien borné $A \subseteq \chi$ et tout $F \in N^{If}$, on définit la grandeur :

$$M(A) = E[N(A)] \text{ et } C(A \times F) = E[N(A) \cdot 1\{X \in F\}]$$

Le lemme suivant donne les conditions sous lesquelles les fonctions ensemblistes $M(\cdot)$ et $C(\cdot, \times)$ peuvent être prolongées en mesures sur la σ -algèbre borélienne $B(\chi)$ de χ et la σ -algèbre produit $B(\chi) \times N^{If}$.

Lemme 2.1 : Si la fonction $M(\cdot)$ introduite dans la définition 2.11 est finie sur les boréliens bornés, alors $M(\cdot)$ peut être prolongée de manière unique en une mesure σ -finie sur les boréliens de χ . On l'appelle alors mesure du moment de premier ordre. La fonction $C(\cdot, \times)$ peut être prolongée de manière unique en une mesure σ finie sur la σ algèbre produit obtenue à partir de $B(\chi)$ et N^{If} . On l'appelle alors la mesure de Campbell de premier ordre.

La mesure du moment de premier ordre peut être exprimée grâce à la mesure de Campbell de la manière suivante :

$$C(A \times N^{If}) = E[N(A)] = M(A)$$

Pour tout borélien borné A . Par conséquent, $C(\cdot)$ est finie si et seulement si $M(\cdot)$ est finie, c'est à dire si et seulement si $E[N(\chi)] < \infty$. Les mesures de Campbell d'ordres supérieurs sont définies suivant le même principe, par exemple :

$$C^{(2)}(A \times B \times F) = E[N(A)N(B).1\{X \in F\}]$$

Les mesures de Campbell sont importantes lorsque l'on travaille avec des processus ponctuels. Supposons que l'on regarde une grandeur $g(x, X)$ en chaque point x d'une réalisation x de X . Par exemple $g(x, x)$ peut être la distance entre x et le point le plus proche de la réalisation x .

Du point de vue pratique, cela est utile pour avoir des critères de convergence pour nos algorithmes. Une autre utilisation de cette mesure concerne la définition de l'intensité conditionnelle d'un processus ponctuel [24].

2.7.2 Lois conditionnelles

Il existe deux façons pour conditionner la loi d'un processus ponctuel:

- Le conditionnement intérieur : on cherche à exprimer la loi conditionnelle d'un processus ponctuel sachant qu'il y a un de ses points en $x \in \chi$. Ce type de conditionnement est formalisé par la théorie de Palm [24].
- Le conditionnement extérieur : on cherche à exprimer la loi conditionnelle d'un point $x \in \chi$ connaissant la configuration sur $\chi \setminus \{x\}$. Ce type de conditionnement est formalisé par la densité conditionnelle de Papangelou.

Le conditionnement intérieur est très intéressant dès lors que l'on dispose de la propriété d'invariance de la loi du processus ponctuel par translation. En particulier il permet de définir et d'utiliser des statistiques spatiales.

CHAPITRE 3

METHODE DE SIMULATION MONTE CARLO PAR CHAINE DE MARKOV

La simulation d'une loi de probabilité ou d'une variable aléatoire est un outil numérique utile chaque fois que l'on ne dispose pas d'une solution analytique à un problème donné, qu'il soit combinatoire ou analytique. L'étude de modèles statistiques complexes a contribué à développer des algorithmes de simulation fiables et efficaces. En particulier la simulation MCMC (Monte Carlo Markov Chain) par l'algorithme de Metropolis-Hastings.

3.1 Méthodes de base de simulation

L'objectif de base des méthodes de simulation est de produire des nombres pseudo aléatoires, ou des variables aléatoires distribuées selon une certaine fonction de densité d'intérêt f . Beaucoup de raisons peuvent pousser à l'utilisation dans la pratique de telles méthodes, par exemple les jeux sur support électronique, ou encore le cryptage de messages, etc. D'un point de vue plus théorique (et c'est à celui-ci que nous nous intéresserons), la simulation servira dans le cadre du problème général de l'évaluation de : $E_f[h(X)] = \int_{\text{sup } f} h(x)f(x)dx$ Pour n'importe quelle variable aléatoire X suivant une fonction de densité f et pour toute fonction réelle h , dans les cas où l'intégrale ne peut être calculée de manière analytique, notamment en grandes dimensions [27].

3.1.1 Intégration de Monte Carlo

La méthode classique d'intégration de Monte Carlo fut développée en 1949 par Stan Ulam et Nicholas Metropolis, avec la collaboration de John Von Neumann, suite au développement du premier ordinateur électronique, ENIAC, qui permettait pour la première fois de l'histoire de se débarrasser des calculs longs et répétitifs. Elle s'énonce comme suit : Supposons que nous ayons un échantillon (X_1, X_2, \dots, X_N) généré selon la densité f (nous nous intéresserons plus loin à la production de cet échantillon). Il est alors possible d'approximer $E_f[h(x)]$ par l'estimateur de Monte Carlo :

$$\bar{h}_N = \frac{1}{N} \sum_{j=1}^N h(x_j)$$

Étant donné que \bar{h}_N converge presque sûrement vers $E_f[h(x)]$ par la loi forte des grands nombres. Bien qu'il soit parfois difficile de produire un échantillon selon f , cette difficulté peut heureusement être contournée par la méthode dite d'échantillonnage préférentiel (en anglais, importance sampling) permettant de générer l'échantillon (X_1, X_2, \dots, X_N) non pas de la distribution d'intérêt f , mais plutôt d'une autre densité g telle que $\text{supp } f \subseteq \text{supp } g$. On utilisera alors le fait que :

$$E_f[h(x)] = \int_{\text{supp } f} h(x) \frac{f(x)}{g(x)} g(x) dx$$

Et $E_f[h(x)]$ sera estimée par :

$$\bar{h}_N = \frac{1}{N} \sum_{j=1}^N \frac{f(x_j)}{g(x_j)} h(x_j)$$

Ainsi, à condition de pouvoir générer l'échantillon selon la distribution voulue, le problème du calcul de $E_f[h(x)]$ est résolu.

3.1.2 Uniformes

Nous pouvons maintenant songer à la production d'un échantillon (X_1, X_2, \dots, X_N) suivant une densité f . Nous aurons avant tout besoin de pouvoir simuler certaines variables aléatoires suivant des distributions simples. Logiquement, la toute première forme de variable aléatoire qu'il serait utile de simuler est sans aucun doute les uniformes sur un intervalle $[a, b]$, et plus spécifiquement les $U_{[a,b]}$ parce qu'elles fournissent une représentation des probabilités et parce que nous devons faire appel aux uniformes pour simuler des variables provenant d'autres distributions. On appelle générateur de nombres uniformes pseudo-aléatoire tout algorithme qui, à partir d'une valeur initiale u_0 et suivant une transformation D , produit une suite $(u_i) = D^i(u_0)$ de valeurs dans $[0, 1]$ satisfaisant la propriété qui suit. Pour n'importe quel n , les valeurs (u_1, \dots, u_n) se comportent, si soumises à certains tests, comme un échantillon (U_1, \dots, U_n) de $U_{[0,1]}$ *i.i.d*

(indépendantes et identiquement distribuées). Un algorithme de base utiliserait par exemple une transformation de la forme $D(x) = \frac{ax+b}{M+1} \bmod 1$, avec $a, b, M \in \mathbb{N}$.

3.1.3 Variables aléatoires discrètes

Grâce à l'utilisation d'uniformes, la simulation de variables aléatoires discrètes devient assez facile. En effet, il suffit de calculer les valeurs que prend la fonction de répartition $F(k) = P(X \leq k)$, p_k pour toutes les valeurs k que peut prendre X . Ensuite, une valeur $u \sim U_{[0,1]}$ est générée. Pour respecter la fonction de répartition F , il suffit donc que X_i prenne la valeur du k tel que $p_{k-1} < u \leq p_k$. En recommençant ces étapes pour i entre 0 et N , on obtient un échantillon (X_1, \dots, X_N) de taille N distribué selon la fonction de masse choisie.

3.1.4 Méthodes de transformations générales

Le problème se complique rapidement lorsque l'on cherche à simuler des variables aléatoires continues. Heureusement, plusieurs se génèrent en utilisant quelques transformations directes à partir d'uniformes. Définissons d'abord, pour une fonction non-décroissante F sur \mathbb{R} (une fonction de répartition par exemple), son inverse généralisée, F^{-1} . Elle sera la fonction définie par

$$F^{-1}(\mu) = \inf \{x : F(x) \geq \mu\}.$$

On pourra utiliser cette inverse pour faire un premier type de transformation. En effet, si $U \sim U_{[0,1]}$, alors la variable aléatoire $F^{-1}(U)$ suit la distribution F . En d'autres termes, pour générer une variable $X \sim F$, il suffit de générer U selon une $U[0,1]$ puis d'effectuer la transformation $x = F^{-1}(\mu)$. Par exemple, pour générer $X \sim \text{Exp}(1)$, donc avec $F(x) = 1 - e^{-x}$, on pourra utiliser $x = -\log(1 - u)$. Si $U \sim U_{[0,1]}$ ($1 - U$ est donc aussi uniforme sur $[0, 1]$), alors la variable $X = -\log U$ aura la distribution exponentielle voulue.

Lorsqu'inverser la fonction de répartition se révèle impossible, nous pourrions utiliser d'autres opérations sur les variables aléatoires pour nous permettre d'en atteindre de nouvelles. Ainsi, en additionnant correctement des exponentielles, ou en utilisant les fonctions trigonométriques, on peut générer, toujours à partir d'uniformes, certains types de variables Chi carrées, Gamma, Beta, Normales, Poisson, etc. Par exemple, l'algorithme suivant, dit Box-Muller, permet de générer des variables suivant des $N(0, 1)$:

- a. Générer U_1, U_2 , deux $U_{[0,1]}$ i.i.d
- b. Définir

$$\begin{cases} x_1 = \sqrt{-2\log(u_1)} \cos(2\pi u_2) \\ x_2 = \sqrt{-2\log(u_1)} \sin(2\pi u_2) \end{cases}$$

- c. Prendre x_1 et x_2 comme étant deux tirages indépendants de $N(0, 1)$.

Il existe bien entendu d'autres méthodes de simulation plus complètes et offrant beaucoup plus de possibilités en termes de densités à simuler. Nommons par exemple la méthode d'acceptation-rejet, qui permet de simuler un échantillon provenant de virtuellement n'importe quelle densité f connue à une constante multiplicative près, en utilisant une seconde densité g , dite instrumentale. Permettons-nous de ne pas développer ici cette méthode, mais de seulement mentionner que le choix de cette seconde densité donne vite beaucoup de fil à retordre, surtout dans les cas de grandes dimensions, et de passer tout de suite à la famille de méthodes qui nous intéresse ici, les méthodes MCMC.

3.2. Chaînes de Markov

On appelle chaîne de Markov à temps discret toute suite de variables aléatoires $(X_N)_{N \in \mathbb{N}}$ à valeurs dans \mathcal{A} , telle que, pour tout entier $k \in \mathbb{N}$ et $A \in \mathcal{A}$, la suite respectant la propriété markovienne suivante :

$$P(X_{N+1} \in A | X_0, X_1, \dots, X_N) = P(X_{N+1} \in A | X_N), \forall A \in \mathcal{A}$$

Autrement dit, la valeur d'une variable aléatoire de cette suite ne dépend que de celle qui la précède. Nous nous intéresserons ici aux chaînes dites homogènes, c'est-à-dire aux chaînes dont son évolution ne dépend pas de la position dans la chaîne, mais seulement de l'état actuel :

$$P(X_{t1}, X_{t2}, \dots, X_{tk} / X_{t0}) = P(X_{t1-t0}, X_{t2-t0}, \dots, X_{tk-t0} / X_0)$$

D'un point de vue informatique, une telle chaîne présente bien entendu l'avantage de rendre inutile le rappel de l'ensemble des configurations antérieures puisqu'elle utilise uniquement l'état actuel pour générer une nouvelle configuration. La génération de la nouvelle configuration réclame la définition d'un noyau de transition. Un noyau de transition est une fonction $P : \mathcal{X} \times \mathcal{A} \rightarrow [0,1]$, tel que :

- pour tout $A \in \mathcal{A}$, $P(\cdot, A)$ est mesurable.

- pour tout $X \in \mathcal{X}$, la fonction $P(X, \cdot)$ est une mesure de probabilité.

Ainsi, une chaîne de Markov homogène est complètement définie par la valeur ou la distribution de X_0 et son noyau de transition P .

Nous pouvons simuler une densité π de façon approchée en utilisant une chaîne de Markov chaque fois qu'on sait construire une transition P telle que, pour toute loi initiale $X_0 = \nu$, $\nu P^k \rightarrow \pi$. Avant de présenter les différents résultats théoriques et algorithmes de simulation, rappelons quelques définitions préliminaires sur les chaînes de Markov.

- **Invariance :**

une loi π est invariante pour la chaîne de Markov si :

$$\pi = \pi P$$

Cette condition est nécessaire pour obtenir la convergence de la chaîne vers π .

- **Réversibilité**

La chaîne est réversible pour π si le noyau de transition P vérifie :

$$\forall A, B \in \mathcal{A}: \int P(x, A)\pi(dx) = \int P(x, B)\pi(dx)$$

Cette condition implique l'invariance pour π , et signifie que sous la distribution stationnaire π la probabilité de passer de A à B est la même que de passer de B à A . La plupart des algorithmes de simulation sont en réalité construits pour produire des chaînes de Markov réversibles.

- **Irréductible**

La chaîne est dite π -irréductible si :

$$\forall x \in \Omega \text{ et } \forall A \in \mathcal{A} \text{ t. q. } \pi(A) > 0, \quad \exists t/P^t(x, A) > 0$$

Cela signifie que la chaîne a une probabilité non nulle d'atteindre en temps fini tout ensemble π -probable. Cette condition est clairement nécessaire pour que la chaîne converge en distribution vers π avec n'importe quelle condition initiale. Dans le cadre de ce projet, nous verrons uniquement des chaînes irréductibles, c'est-à-dire des chaînes ne possédant qu'une seule classe d'états.

- **A périodicité**

A périodicité assure que les déplacements entre états n'ont pas trop de contraintes. Formellement, la chaîne est apériodique s'il n'existe de partition disjointe $A = \bigcup_{i=0}^k A_i$ pour $r \geq 2$ tel que :

$$P(x, A_i) = 1, \forall x \in A_i$$

Il existe un lien entre l'irréductibilité et l'apériodicité, dans la mesure où si une chaîne de Markov est irréductible, et s'il existe un état X_k tel que la densité $P(X_k, X_k) > 0$, la chaîne de Markov est fortement apériodique. Donc tous les processus pour lesquels il est possible de rester dans un état sont fortement apériodiques (c'est le cas des algorithmes de type Metropolis-Hasting que nous étudierons plus loin).

- Une chaîne π -irréductible et π -invariante est récurrente positive si, $\forall A \in \mathcal{A}$ t.q. $\pi(A) > 0$, elle vérifie :

$$\forall x, P_x\{X \in A \text{ i. s}\} > 0 \text{ et } P_x\{X \in A \text{ i. s}\} = 1$$

(i.s : infiniment souvent)

- Sur un espace discret le noyau de transition P est primitif (régulier) si $\exists k \geq 1$ tel que P^k a tous ses termes strictement positifs.

3.2.1 Convergence d'une chaîne de Markov

Avant de présenter les méthodes de simulation des chaînes MCMC, nous présentons les conditions nécessaires pour qu'une chaîne puisse converger et atteindre la distribution recherchée π . Si, P est π -irréductible, π -invariante, récurrente. Dans ce cas, la mesure invariante est unique et la chaîne est dite récurrente positive si la masse totale de cette mesure est finie. C'est le cas si π est une probabilité. On obtient les résultats d'ergodicité [28] :

Proposition 3.1 Si, P est π -irréductible, P est π -invariante, alors P est récurrente positive et π est l'unique loi invariante de P . Si P est apériodique, alors $\forall \chi$, on a :

$$\|P^m(x, \cdot) - \pi\| \rightarrow 0 \text{ quand } m \rightarrow \infty$$

Le contrôle de la convergence vers 0 de $\|P^m(x, \cdot) - \pi\| \rightarrow 0$ est une question centrale et très difficile. Ce contrôle permet en effet d'assurer que vP^m pour m assez grand réalise une simulation acceptable de π . Il existe de nombreux résultats théoriques. Certains utilisent des minoration de P sur un petit ensemble E . Le coefficient de contraction sur un espace d'états fini permet le contrôle de cette convergence [29].

- **Ergodicité géométrique, ergodicité uniforme**

L'ergodicité géométrique est caractérisée par :

$$\|P^m(x, \cdot) - \pi\| \leq M(x) l^m$$

Où $M(x)$ est π - intégrable et $l < 1$. L'ergodicité est uniforme si on peut choisir pour M une constante finie.

- **Coefficient de contraction**

Dans un espace d'états fini, le coefficient de contraction pour un noyau de transition P est donné par [30] :

$$C(P) = \frac{1}{2} \max_{x,y \in E} \|P(x, \cdot) - P(y, \cdot)\|$$

Lemme 3.2 [31].

Soient ν et μ deux distributions, P et Q deux noyaux de transition. Alors

$$\|\mu P - \nu P\| \leq \|\mu - \nu\| C(P) \text{ et } C(PQ) \leq C(P)C(Q)$$

En particulier,

$$\|\mu P - \nu P\| \leq \|\mu - \nu\| \text{ et } \|\mu P - \nu P\| \leq 2C(P)$$

Et si P est primitif alors $C(P) < 1$.

D'après ce de dernier résultat, si nous prenons $\mu = \pi$, la loi invariante de P , on en déduit :

$$\|\nu P^m - \pi P^m\| = \|\nu P^m - \pi\| \leq 2C(P^m) \leq 2C(P)^m$$

Dans ce cas, quand $m \rightarrow \infty$ la chaîne est uniformément ergodique si P est primitif.

Signalons à la fin de cette section, que ces dernières conditions, délicates à obtenir en général, sont satisfaites pour les deux principales familles de simulateurs : l'algorithme de Metropolis-Hastings et l'algorithme de Gibbs, l'objectif de la section suivante.

3.3. La méthode Monte Carlo par chaînes de Markov

Les méthodes d'échantillonnage sont utilisées pour estimer la distribution (complète) postérieure des paramètres d'un modèle. Parmi elles, les méthodes de Monte Carlo sont très précises. Néanmoins, elles sont coûteuses en calculs et prennent beaucoup de temps à converger. Elles sont également limitées par la taille de l'échantillon, puisqu'elles deviennent insolubles lorsque ceux-ci sont trop grands. Même sur de petits échantillons, le calcul d'une distribution de probabilité peut s'avérer difficile. Il existe plusieurs approches à

ce problème, utilisées pour obtenir un ensemble de bons réseaux d'échantillonnage à partir de la distribution postérieure.

Définition 3.1

Nous appellerons MCMC toute méthode permettant de simuler une distribution en utilisant une chaîne de Markov ergodique $(X_N)_{N \in \mathbb{N}}$ ayant celle-ci comme distribution stationnaire

3.3.1 La base de la méthode

La méthode Monte Carlo par chaînes de Markov permet l'estimation des densités de probabilité a posteriori pour des modèles multidimensionnels et fournit des estimations fiables des erreurs. Les algorithmes MCMC nous permettent d'échantillonner à partir d'une distribution de probabilité connue à une constante multiplicative près. Ces algorithmes consistent à simuler séquentiellement une seule chaîne de Markov dont la distribution limite est celle choisie (dans notre cas, le maximum de la fonction de vraisemblance fois une densité de probabilité a priori des paramètres). Plus précisément, une chaîne de Markov est une séquence de variables aléatoires telle que la valeur suivante ou état de la séquence dépend uniquement de l'état présent et non des états passés (absence de mémoire).

CHAPITRE 4

ETUDE COMPARATIVE ENTRE LES DEUX ALGORITHMES DE METROPOLIS-HASTING STANDARD ET A RETARD REJETE

Dans ce chapitre nous nous intéressons en premier temps, à l'étude des deux algorithmes de Métropolis-Hasting : l'Algorithme sous sa forme générale et à retard rejeté. Nous introduisons en suite le processus markovien de Strauss Marqués utilisé pour la génération des matrices d'expériences numériques de Strauss. Dans un deuxième temps, nous fournissons une comparaison entre les deux versions de l'algorithme. Cette comparaison a été réalisée en développant un programme dans le logiciel Matlab présenté en annexe de ce manuscrit.

4.1 Méthode de l'échantillonnage

Les méthodes de Monte Carlo par chaînes de Markov (MCMC), permettent de simuler une distribution en utilisant une chaîne de Markov ergodique ayant celle-ci comme distribution stationnaire. Pour construire un tel algorithme, il faut donc déterminer un ensemble de probabilités de transition P approprié, c'est-à-dire irréductible, ergodique et ayant la bonne distribution stationnaire.

4.2 L'algorithme de Metropolis-Hastings Généralisé

L'algorithme Metropolis a été introduit par Metropolis, et al. en 1953 [32] et généralisé par Hastings en 1970 [4], il a été adapté pour le cas des processus spatiaux par Geyer et Moller 1994[33]. L'idée de base de l'algorithme est de proposer un nouvel état par une petite perturbation de l'état actuel, puis de l'accepter ou le refuser. Il est le cheval de bataille standard de calcul des méthodes MCMC tant pour sa simplicité et pour sa polyvalence, et est en principe applicable à tout système. Il est extrêmement facile à implémenter pour échantillonner une densité cible $\pi(x)$.

Le taux d'acceptation $R(x, y)$, représente la probabilité de transition de la chaîne d'un état actuel x vers l'état y . L'algorithme proposé par Geyer et Moller construit une chaîne de Markov qui peut explorer toutes les configurations de l'espace. Les perturbations

proposées sont la naissance et la mort (ajout ou suppression d'un élément de la configuration actuelle). En général, l'algorithme utilise une transition P qui est π -réversible et π invariante. La construction de P se fait en deux étapes :

- Transition de proposition de changement : on commence par proposer un changement $x \rightarrow y$ avec une probabilité $Q(x, dy)$.
- Probabilité d'acceptation du changement : ensuite, on accepte ce changement avec la probabilité $a(x, y)$ où $a : \Omega \times \Omega \rightarrow]0,1]$. Si non on reste en x .

Les deux paramètres de l'algorithme sont Q , la transition de proposition de changement et a la probabilité d'accepter ce changement. Si $\delta_x(dy)$ est la mesure de Dirac en x la transition P de MH s'écrit :

$$P(x, dy) = a(x, y)Q(x, dy) + \delta_x(dy) \int_{\Omega} 1 - a(x, z)Q(x, dz)dz$$

Ecrire que P est π -réversible impose une contrainte sur Q et la probabilité d'acceptation a . Notons que : $\nu(dx, dy) = \mu(dx, dy) + \mu(dy, dx)$, où $\mu(dx, dy) = Q(x, dy)\pi(dx)$ et μ admet une densité $h(x, y)$ par rapport à ν . Le ratio r de Metropolis-Hastings est défini par :

$$r(x, y) = \frac{h(x, y)}{h(y, x)} \text{ sur } R = \{(x, y) \mid h(x, y) > 0 \text{ et } h(y, x) > 0\}$$

La transition P est π -réversible si et seulement si la probabilité d'acceptation a vérifie [34]:

$$a(x, y)r(x, y) = a(y, x)$$

Si, Q et π sont respectivement à densité q et π , et si $q(x, y) > 0$ équivaut $q(y, x) > 0$, alors la π -réversibilité de P s'écrit alors :

$$\forall x, y \in \Omega : \pi(x) \times q(x, y) \times a(x, y) = \pi(y) \times q(y, x) \times a(y, x)$$

La chaîne de Markov s'itère en « sautant » de l'état actuel X_t de l'espace des paramètres à l'état suivant X_{t+1} . La probabilité de transition entre deux états est spécifiée par l'algorithme de Metropolis-Hastings en utilisant une fonction d'échantillonnage $\pi(\cdot | X_t)$.

À chaque temps Monte Carlo t , correspondant à l'état X_t , un état Y_t est généré à partir de la fonction d'échantillonnage $\pi(\cdot | X_t)$ qui a les mêmes propriétés que la distribution cible. Cet état proposé est accepté comme nouvel état de la chaîne avec la probabilité α

Algorithme de Metropolis-Hasting Standard

Données Densité cible π , densité instrumentale q (Dans notre cas la loi uniforme) et la taille de l'échantillon Monte Carlo N .

Procédure

1. Initialisation. Valeur initiale de la chaîne x_0
2. Pour $n = 0, \dots, N - 1$

Proposition. Générer le premier candidat $y \sim q(x_t, \cdot)$

Acceptation. Avec probabilité $\alpha(x_t, y_{t+1}) = \min \left[\frac{\pi(y_{t+1})q(y_{t+1}, x_t)}{\pi(x_t)q(x_t, y_{t+1})}, 1 \right]$

Autrement dit

$$x_{t+1} = \begin{cases} y_{t+1} & \text{avec probabilité } \alpha \\ x_t & \text{avec probabilité } 1 - \alpha \end{cases}$$

$x_{0:N}$

Sortie L'échantillon

La probabilité de transition $T(X_t \rightarrow Y_t)$ prend alors la forme explicite suivante

$$T(X_t \rightarrow Y_t) = \alpha(X_t, Y_t) \pi(Y_t | X_t)$$

Si le nouvel ensemble de paramètres Y_t est accepté, alors il devient le nouvel état de la chaîne $X_{t+1} = Y_t$ (la chaîne « évolue »). S'il est rejeté, alors l'ancien état est réécrit dans la chaîne $X_{t+1} = X_t$ (la chaîne « stagne »). Ce critère assure l'ergodicité de la chaîne, i.e. que la chaîne échantillonne la distribution voulue $\pi(x)$ une fois à l'équilibre. Si notre distribution cible apparaît avec une constante de normalisation que nous ne connaissons pas, alors le calcul du rapport $r(X_t, Y_t) = \left[\frac{\pi(y_t)q(y_t, x_t)}{\pi(x_t)q(x_t, y_t)} \right]$, entraînera la

simplification de ce facteur : c'est l'une des caractéristiques attrayantes de l'algorithme de Metropolis-Hastings, le faisant approprié pour une grande variété d'applications.

Un cas spécial de l'algorithme de Metropolis-Hastings est l'algorithme de Metropolis à marche aléatoire, pour lequel la distribution d'échantillonnage est choisie de sorte que $\pi(Y_t | X_t) = \pi(|X_t - Y_t|)$. La probabilité d'acceptation se réduit donc à

$$r(X_t, Y_t) = \min \left[\frac{\pi(y_t)}{\pi(x_t)}, 1 \right]$$

Fréquemment, la proposition du saut de la marche aléatoire a une forme qui dépend d'un paramètre d'échelle. Par exemple, la fonction de proposition pour Y_t pourrait être une distribution Gaussienne centrée en X_t et d'écart-type σ . Une attention méticuleuse doit être portée au choix de ce paramètre d'échelle. Si ce paramètre d'échelle est très grand, un pourcentage très élevé des points de la chaîne sera rejeté, conduisant à un algorithme inefficace. S'il est très petit, la marche aléatoire explorera très lentement l'espace des paramètres, ce qui conduira encore à l'inefficacité de l'algorithme. Ce problème est d'autant plus difficile à gérer que le nombre de paramètres (i.e. la dimension du vecteur X) à échantillonner augmente.

4.3 Algorithme Metropolis-Hastings avec rejet retardé

L'algorithme Metropolis-Hastings propose un seul candidat à chaque itération. Une extension possible de cet algorithme est de considérer plutôt un ensemble de candidats puis de sélectionner un des candidats les plus intéressants. L'algorithme MH à rejet retardé (DR : Delayed Rejection) [5] est un exemple d'algorithme qui considère plusieurs candidats à chaque itération. Un nombre maximal de candidats K est choisi ; à chaque itération, un candidat est proposé soit jusqu'à ce qu'il soit accepté, soit jusqu'à ce que le nombre maximal soit atteint. Dans le second cas, la chaîne reste sur place tel un rejet MH. La manière séquentielle de produire les candidats fait en sorte que la probabilité d'acceptation M.-H. Des candidats supplémentaires doit être adaptée afin de prendre en compte la conditionnalité sur les candidats précédemment rejetés. Par exemple, pour $K = 2$, on trouve que la probabilité d'acceptation du premier candidat y_1 est la même que celle d'un algorithme MH, puis que celle du second candidat y_2 est donnée par :

$$\alpha_2(x, y_1, y_2) = \min \left(\frac{\pi(y_2)q_1(y_2, y_1)q_2(y_2, y_1, x)[1 - \alpha_1(y_2, y_1)]}{\pi(x)q_1(x, y_1)q_2(x, y_1, y_2)[1 - \alpha_1(x, y_1)]}, 1 \right) \quad (1)$$

et q_2 sont les densités instrumentales respectivement du premier et du second candidat.

Algorithme Metropolis-Hastings à rejet retardé (DR) à deux étapes

Données Densité cible π , densités instrumentales q_1 et q_2 (dans notre cas la lois Normale centrée réduite) et la taille de l'échantillon Monte Carlo N .

Procédure

1. Initialisation. Valeur initiale de la chaîne x_0
2. Pour $n = 0, \dots, N-1$
 - a) Proposition 1. Générer le premier candidat $y_1 \square q_1(\cdot)$
 - b) Acceptation 1. Avec probabilité

$$\alpha_1(x, y_1) = \min \left[\frac{\pi(y_1)q_1(y_1, x)}{\pi(x)q_1(x, y_1)}, 1 \right]$$

accepter la proposition $X_{t+1} = y_1$ et avec probabilité α_1 .passer à l'itération $n + 1$; sinon rejeter la proposition et :

- i. Proposition 2. Générer le second candidat $y_2 \square q_1(x, y_1, \cdot)$
- ii. Acceptation 1. Avec probabilité $\alpha_2(x, y_1, y_2)$ donnée par (1)

$$\text{Et prendre } X_{t+1} = \begin{cases} y_2 & \text{avec probabilité } \alpha_2 \\ x & \text{avec probabilité } 1 - \alpha_2 \end{cases}$$

Sortie L'échantillon $x_{0:N}$.

4.4 Implémentation et comparaison

4.4.1 Matlab comme outil de programmation

Matlab (Matrix laboratory) est un langage de calcul scientifique très performant intégrant le calcul, la programmation et la visualisation dans un environnement simple à

utiliser. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant calcul numérique et visualisation. C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité. MATLAB possède ses données avec d'autres propres langages, intuitifs et naturels qui permettent des gains de temps de CPU spectaculaires par rapport à des langages comme le C, le Turbo-Pascal et le Fortran. Avec MATLAB, on peut faire des liaisons de façon dynamique, à des programmes C ou Fortran, échanger des applications (via la DDE : MATLAB serveur ou client) ou utiliser MATLAB comme moteur d'analyse et de visualisation.

4.4.2 Quelles sont les particularités de MATLAB ?

MATLAB permet le travail interactif soit en mode commande, soit en mode programmation, tout en ayant toujours la possibilité de faire des visualisations graphiques. Considéré comme un des meilleurs langages de programmations (C ou Fortran), MATLAB possède les particularités suivantes par rapport à ces langages :

- La programmation facile,
- La continuité parmi les valeurs entières, réelles et complexes,
- La gamme étendue des nombres et leurs précisions,
- La bibliothèque mathématique très compréhensive,
- L'outil graphique qui inclut les fonctions d'interface graphique et les utilitaires,
- La possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).

4.4.2 Le processus de Strauss Marqué

Dans cette partie, nous allons utiliser le processus de Strauss Marqué dans l'exécution des deux algorithmes précédemment décrits. La densité de probabilité d'un processus ponctuel marqué de Strauss pour une configuration x de points est donnée par [3]:

$$\pi(x) = \alpha \beta^{n(x)} \gamma^{s(x)}$$

Avec,

- α est la constante de normalisation,

- $0 < \gamma < 1$ est un coefficient d'interaction,
- $\beta > 0$ est la densité du processus,
- $s(x) = \sum_{i < j} 1_{\{x_i \sim y_j\}}$ le nombre de paires de voisins par rapport à la relation \sim . Il s'agit simplement du nombre de paires d'expériences (x_i, x_j) telles que la distance entre x_i et x_j est inférieure ou égale à r . Par conséquent $s(x) = \sum_{i < j} 1_{\|x_i - x_j\| \leq r}$.
- $n(x) = \sum_{i=1}^M n(x \times \{i\})$, $n(x \times \{i\})$ représente le nombre de points dans la configuration x ayant la marque i .

Pour le choix des marques, nous appuyons sur le même choix utilisé par Elmoosaoui *et al* [2, 3].

4.4.3 Résultats de la Comparaison

Pour juger la qualité d'un plan d'expériences numérique, il est important d'utiliser des critères usuels permettant d'assurer un bon remplissage de l'espace et une bonne distribution uniforme. L'objectif de cette section, est de calculer les valeurs de ces critères présentés dans le premier chapitre, sur les plans de Strauss, générés à l'aide des deux versions de l'algorithme de Métropolis-Hasting. Pour cela, nous utilisons trois types de critères :

- Critère de distance,
- Critère de recouvrement,
- Critère de discrédance.

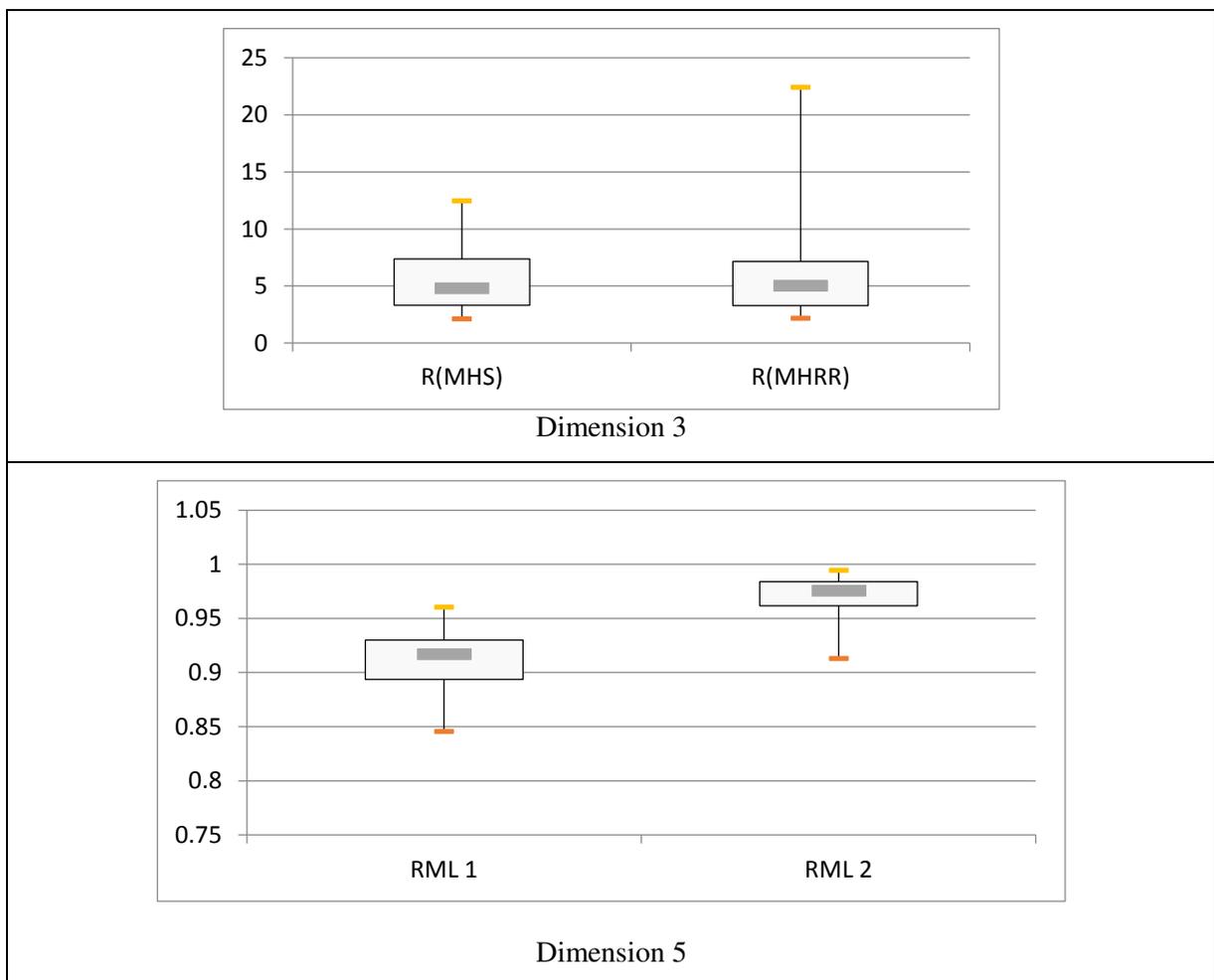
Afin de comparer les deux approches en dimension supérieure à deux, nous rappelons que :

- Le critère de distance consiste à maximiser la distance minimale entre deux points du plan. Plus la valeur de ce critère est grande, plus les points seront éloignés les uns des autres,
- Le critère de recouvrement (Cov) permet de mesurer l'écart entre les points du plan et ceux d'une grille régulière. Ce critère est nul pour une grille régulière. Le but est donc de minimiser le recouvrement pour se rapprocher d'une grille régulière, et ainsi assurer le remplissage de l'espace,
- Critère de discrédance : la discrédance permet de mesurer l'écart entre la fonction de répartition empirique des points du plan et celle de la loi uniforme. Plus la

discrédance est faible, plus les points sont répartis uniformément. Il existe différentes mesures de discrédance. Nous retenons la discrédance en norme L2.

Afin de donner un sens aux résultats. Les figures ci-dessous représentent les box-plots obtenus à la suite de cette étude pour les deux versions de l'algorithme étudiées et pour plusieurs dimensions. Nous avons aussi choisi la loi Normale centrée réduite comme moteur de génération des variables aléatoire pour les deux versions, ainsi pour les instrumentales q_1 et q_2 .

4.4.3.1 Résultat le critère R



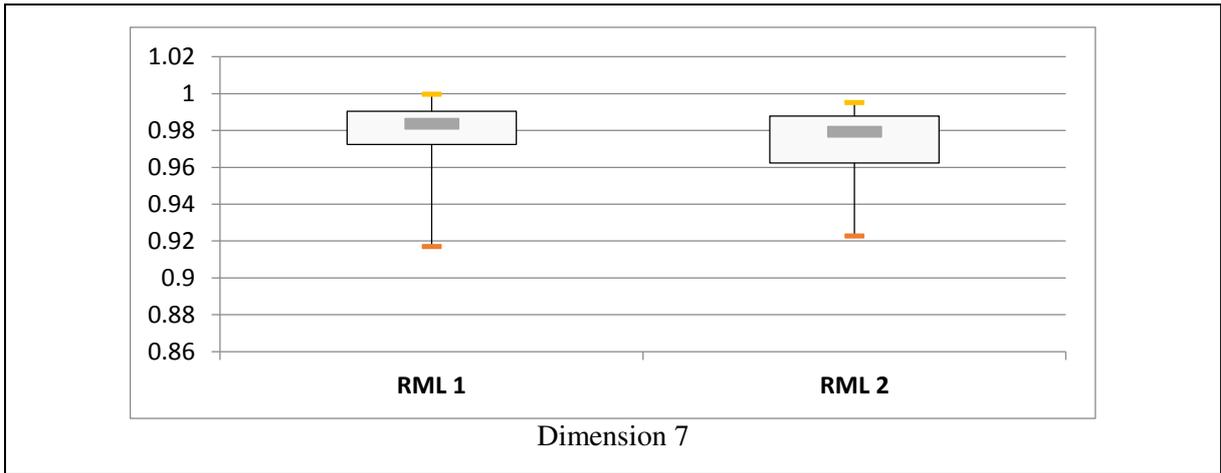
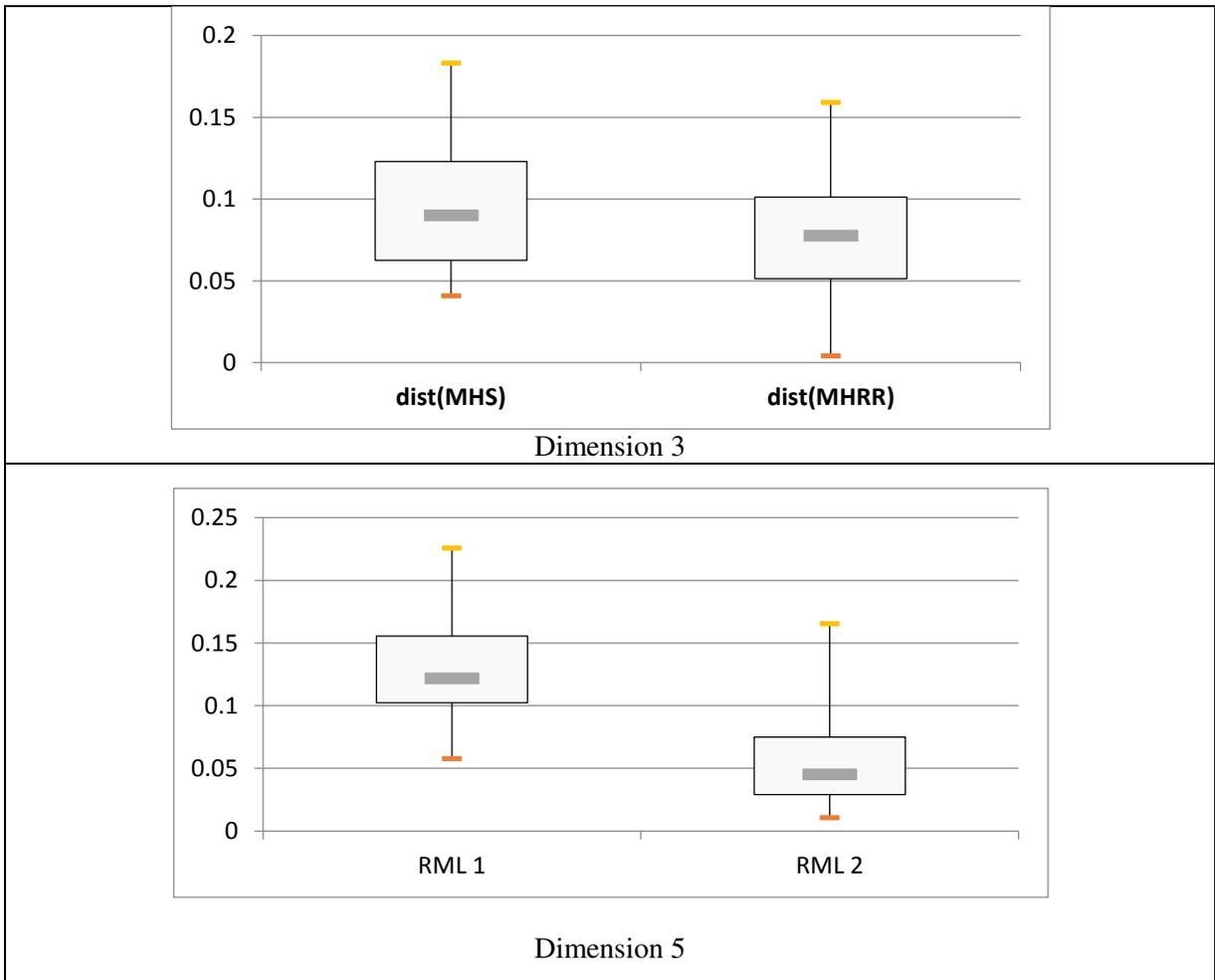


Figure 4.1 : Boxplots de critère de qualité R, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions.

Selon les résultats illustrés par la figure ci-dessus, on remarque que la meilleure valeur de critère R est obtenue par l’algorithme standard de Métropolis-Hasting, sauf pour le cas où la dimension est 3.

4.4.3.2 Comparaison selon le critère de Distance



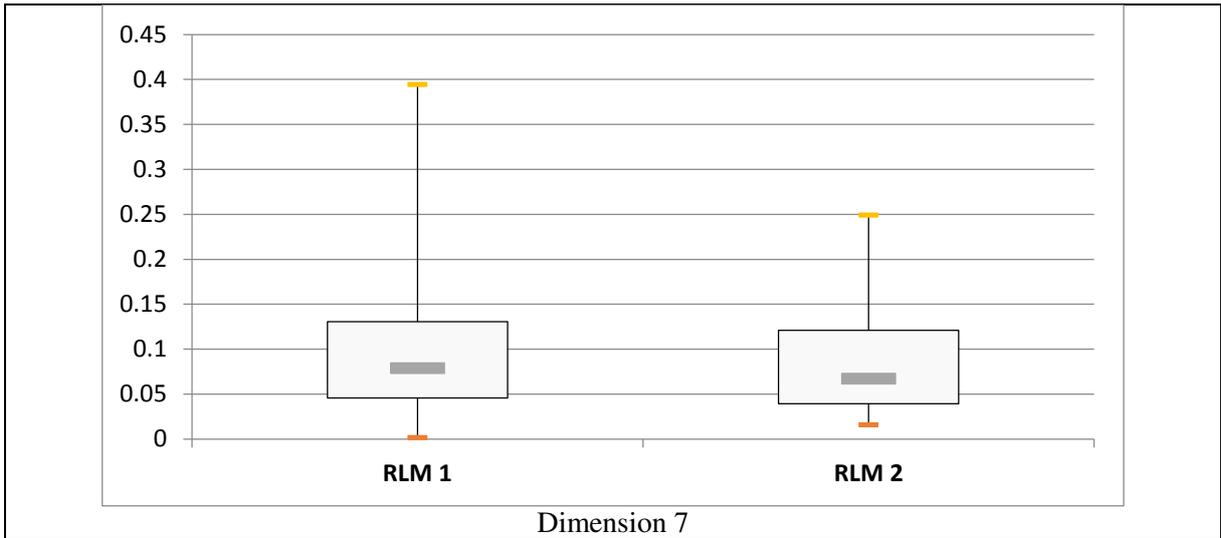
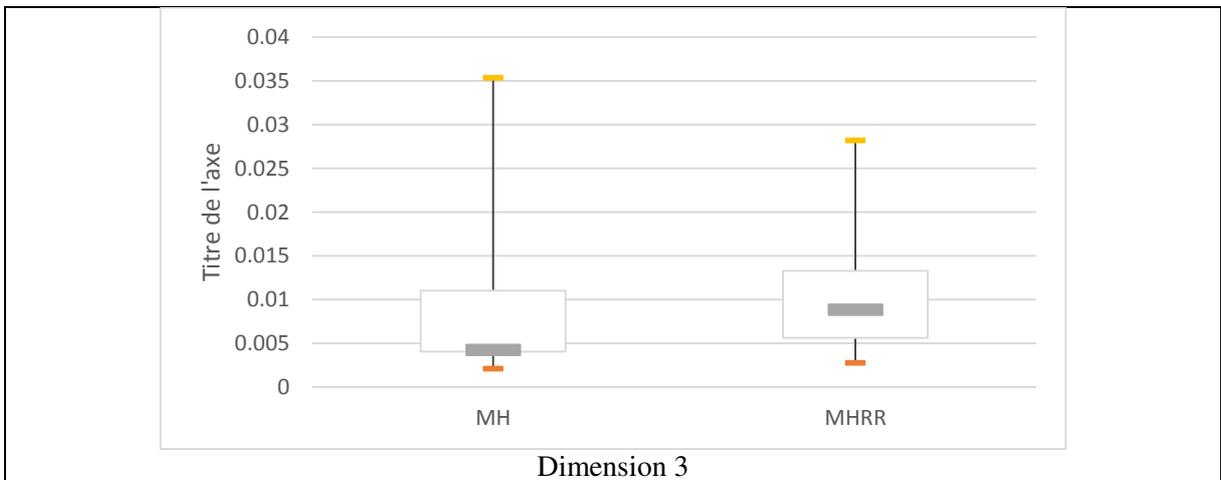


Figure 4.4 : Boxplots de critère de qualité de distance, calculé sur 40 plans à 30 points en dimension 3, à 40 points pour dimension 5 et à 50 points pour dimension 7.

Selon les résultats illustrés par la figure ci-dessus, on remarque que la meilleure valeur de critère de distance est obtenue par l’algorithme de Métropolis-Hasting à retard rejeté.

4.4.3.3 Comparaison selon le critère de discrèpence



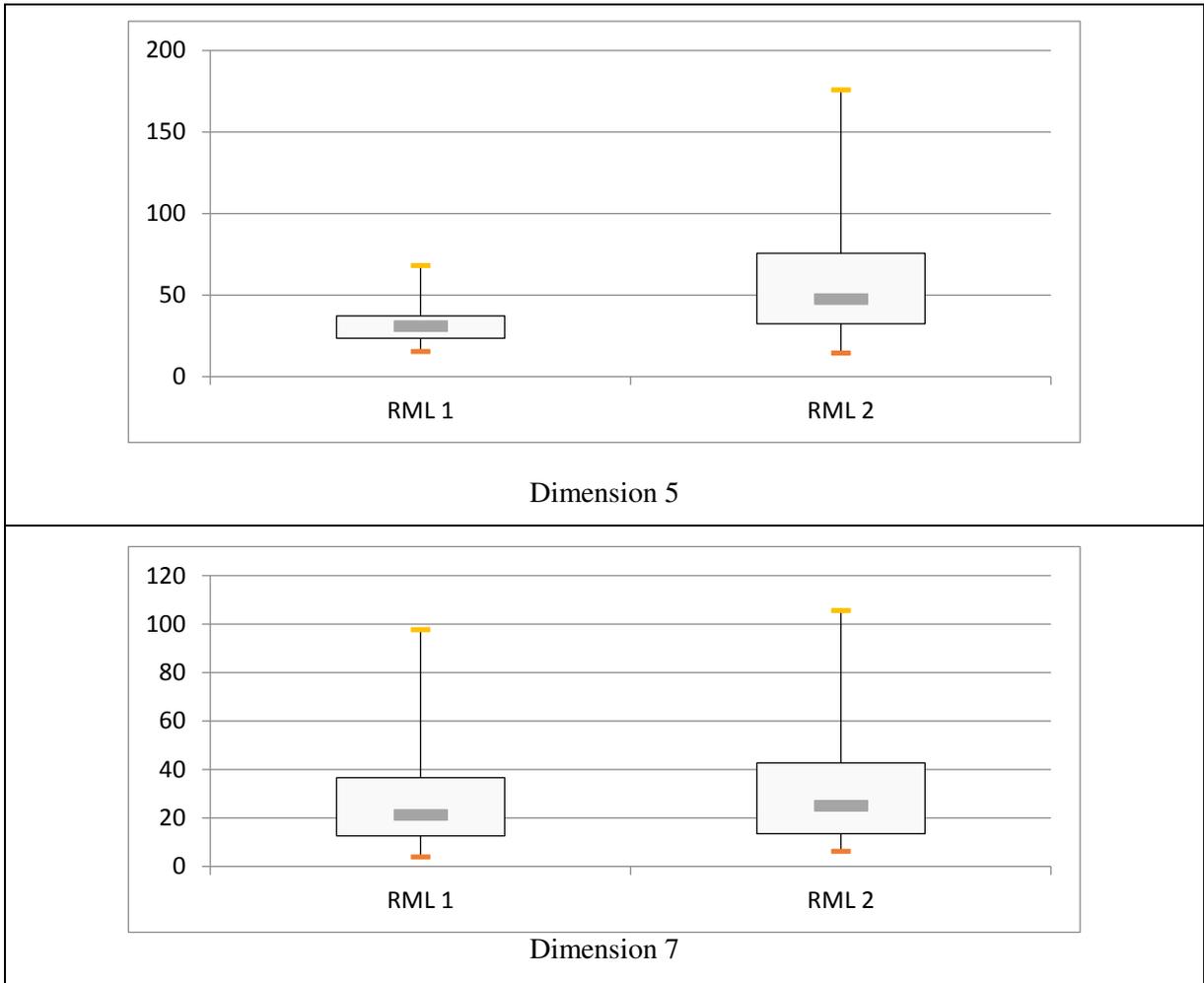
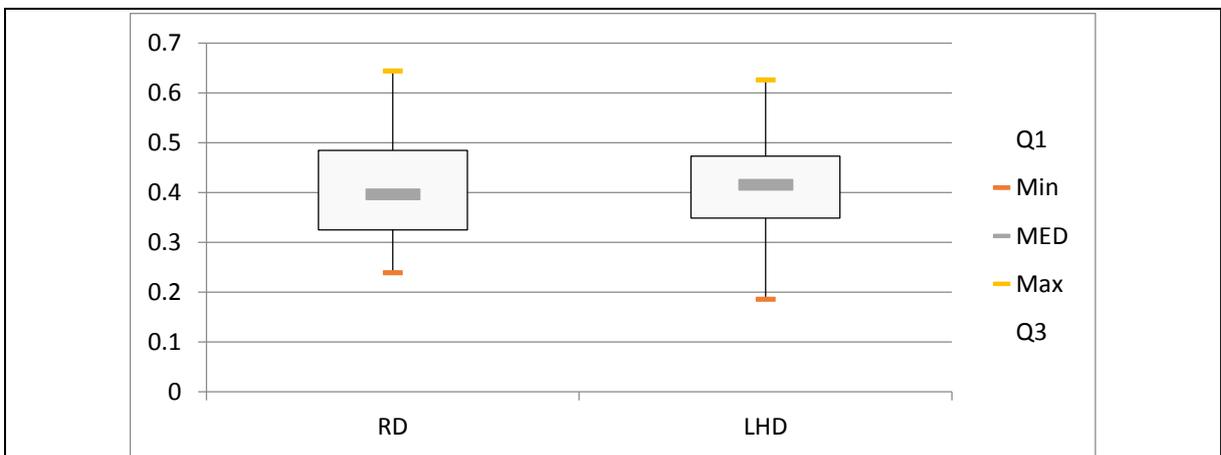


Figure 4.3 : Boxplots de critère de discrèpence, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions.

Selon les résultats illustrés par la figure ci-dessus, on remarque que la meilleure valeur de critère de distance est obtenue par l’algorithme général de Métropolis-Hasting standard.

4.4.3.4 Comparaison selon le critère de couverture



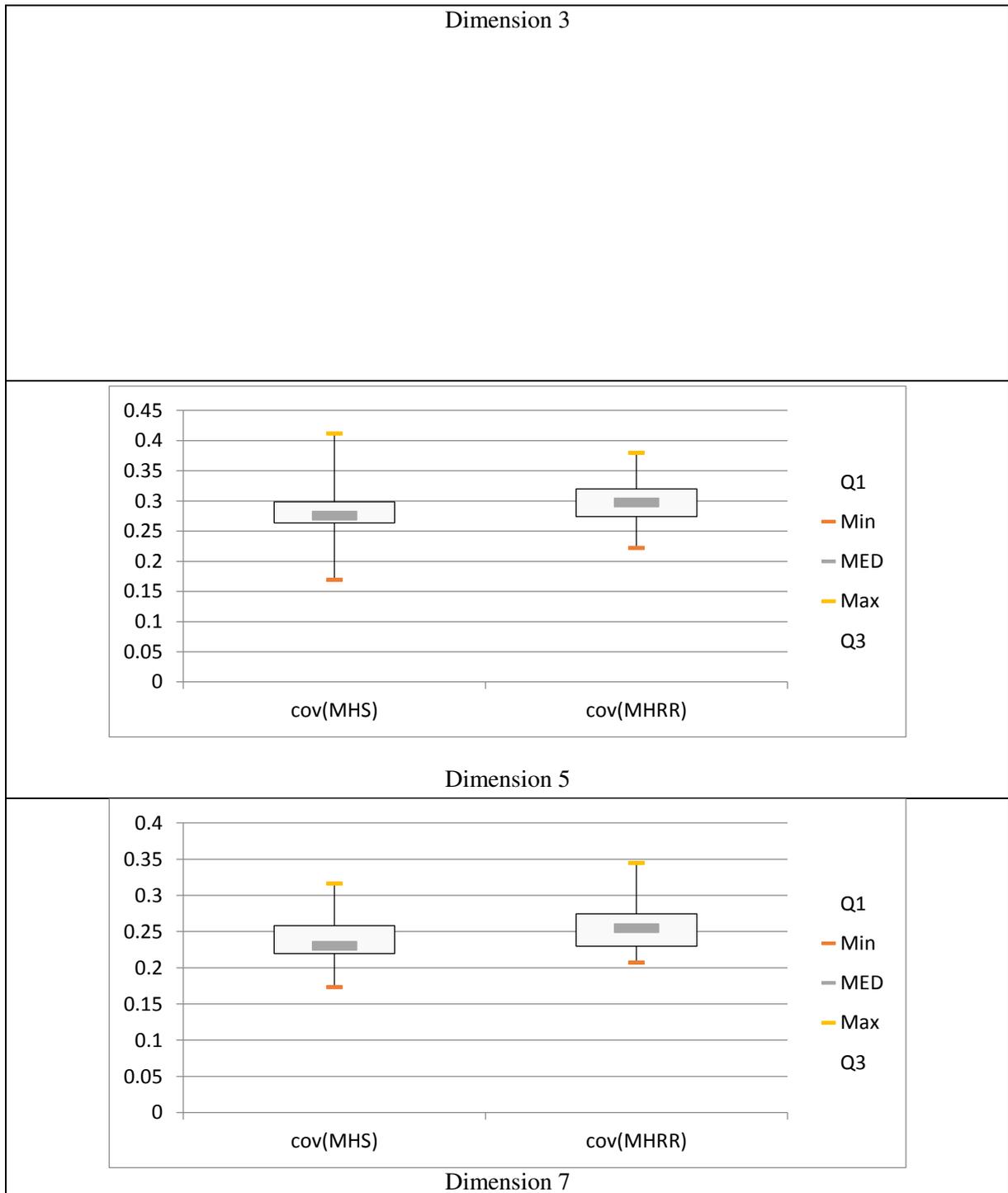


Figure 4.4 : Boxplots de critère de qualité recouvrement, calculé sur 40 plans à 30 points en 3 dimensions, à 40 points pour 5 dimensions et à 50 points pour 7 dimensions.

Selon les résultats illustrés par la figure ci-dessus, on remarque que la meilleure valeur de critère de distance est obtenue par l'algorithme général de Métropolis-Hasting standard.

Une analyse rapide des résultats présentés par les quatre figures ci-dessus, montre que dans la majorité des cas, l'algorithme de Métropolis Hasting standard donne les meilleurs résultats vis à vis des quatre critères misent en jeu dans cette comparaison. Cette constatation, peut être due au choix des densités instrumentales pour q_1 et q_2 .

CONCLUSION

La méthode des plans d'expériences est un ensemble de méthode et de mode de raisonnement destiné à tout expérimentateur désirant faire de la planification expérimentale, dans ce contexte nous nous sommes intéressés à la construction des plans d'expériences à partir de simulateurs déterministes et en phase exploratoire i.e. lorsque la dépendance entre les entrées et les sorties est à priori inconnue.

L'utilisation des méthodes MCMC et l'algorithme de Metropolis Hasting dans le contexte des plans d'expériences numériques permet de construire de nouveaux plans spécifiés à partir d'une certaine loi. Cette approche offre une grande souplesse dans la mesure où l'on peut facilement jouer sur cette loi via sa représentation pour imposer des propriétés comme le remplissage du domaine d'étude par exemple.

Dans ce travail, nous nous sommes intéressés en premier lieu à l'étude de la théorie de processus ponctuels et la méthode MCMC utilisées dans la génération de matrices d'expériences numériques.

En second lieu, une étude et une comparaison ont été réalisées concernant les deux algorithmes de Metropolis-Hasting: L'algorithme de Métropolis-Hasting sous sa forme général et l'algorithme de Metropolis-Hastings à retardé rejeté.

En fin, à travers de la comparaison effectuée dans le cadre de ce travail, nous nous avons pas remarqué une grande efficacité remarquable de l'utilisation de l'algorithme de Metropolis-Hasting avec rejet retardé (à essais multiples), cela, peut être dû au choix de la densité instrumentale.

APPENDICE A

CODE MATLAB POUR LES RESULTATS PRESENTES DANS LE CHAPITRE 4

```
% % % % % % % % % % % % % % % % % % % % % % % % % % % % %  
% % % % % %
```

Code Matlab pour le critère de déscrepance

```
%la discrepance  
function y=dsc(X)  
n=20; w=3;  
s1=0;  
for i=1:n  
    p1=1;  
    for j=1:w  
        p1=p1*(1-X(i,j))*(1+X(i,j));  
    end  
    s1=s1+p1;  
end  
s2=0;  
for i=1:n  
    for j=1:n  
        p2=1;  
        for k=1:w  
            m=max(X(i,k),X(j,k));  
            p2=p2*(1-m);  
        end  
        s2=s2+p2;  
    end  
end  
y=(3^(-w))+1/(n^2)*s2-(1/(n*2^(w-1)))*s1;  
end  
% % % % % % % % % % % % % % % % % % % % % % % % % % % % %  
% % % % % %
```

Code Matlab pour le critère de recouvrement

```
%le critere de recouvrement
function y=mindist(X)
n=20; w=3;
for i=1:n-1
    for k=i+1:n
        s=0;
        for j=1:w
            s=s+((X(i,j)-X(k,j))^2);
        end
        M(i,k)=s^0.5;
        M(k,i)=s^0.5;
    end
end
for i=1:n
    M(i,i)=inf;
end
for i=1:n
    t(i)=min(M(i,:));
end
q=0;
for i=1:n
    q=q+t(i);
end
q1=q/n;
lamda=0;
for i=1:n
    lamda=lamda+(t(i)-q1)^2;
end
y=(1/q1)*((1/n)*(lamda))^0.5;
end
```



```

Sy1=0;
for i=1:n
for j=i+1:n
for k=1:w
d=y1(i,k)-y1(j,k);
end
end
Dy=norm(d);
if(Dy<=R)
Sy1=Sy1+1;
else
Sy1=Sy1;
end
end
end

```

```

SX=0;
for i=1:n
for j=i+1:n
for k=1:w
d(k)=X(i,k)-X(j,k);
end
D=norm(d);
if(D<=R)
SX=SX+1;

```

```

end
end

```

```

end
% calculate a prob d'acc
by1=t^Sy1;
bX=t^SX;

```

```

a=min(1,by1/bX);
if (a==1)
X=y1;

```

```

else
for i=1:n
for j=1:w
test=0;
while (test==0)
z=randn;
if ((z > 0)&&(z <1))
test=1;
end
end
end
y2(i,j)=z;

```

```

end
end
Z=(X+y1)/2;

for i=1:n
for j=1:w
N=y2(i,j)*0.05+Z(i,j);
if ((N > 0)&&(N < 1))
y2(i,j)=N;
end
end
end
Sy2=0;
for i=1:n
for j=i+1:n
for k=1:w
d=y2(i,k)-y2(j,k);
end
Dy=norm(d);
if(Dy<=R)
Sy2=Sy2+1;
else
Sy2=Sy2;
end

end
by2=t^Sy2;
bX=t^SX;
by1=t^Sy1;
a1=min(1,by1/by2);

a2=min(1,(by2)*(1-a1)/(bX)*(1-a));
if (a2==1)
X=y2;

end
end

v(nbrplan,1)=rap(X);
% Rap=Rap+rb;
%resultt(nbrplan,1)=rb;

v(nbrplan,2)=dsc(X);
%Desc=Desc+dc;
%resultt(nbrplan,2)=dc;

v(nbrplan,3)=mdist(X);

```

```
% Mind=Mind+mn;  
%resultt(nbrplan,3)=mn;  
v(nbrplan,4)=mindist(X);  
  
end  
end  
end  
  
xlswrite('bprof2.xlsx',v);
```

REFERENCES BIBLIOGRAPHIQUES

1. Goupy, J., 2017. , ”*Introduction Aux Plans D'experiences*“. Paris: Dunod.
2. Myers, R.H. and Montgomery, D.C. Response Surface Methodology: “Process and Product Optimization Using Designed Experiments“. John Wiley and Sons, New York(1995)
3. Dreesbeske, J, Fine, J et Saporta, G, “Plan d’expérience Applications a l’entreprise“ (1997)
4. YAHIAOUI.I, “ thèse de doctorat, Les plans d’expériences Initiation à la construction et l’analyse des plans factoriels complets à deux niveaux et du plan composite centré “, Université A. MIRA – Bejaia
5. J.FRANCO, “thèse de doctorat, “ Planification d’expériences numériques en phase exploratoire pour la simulation des phénomènes complexes, u département Méthodes et Modèles Mathématiques pour l’Industrie (3MI) de l’ENSM-SE et au département Géomodélisation et Incertitudes“, Ecole Nationale Supérieure des Mines de Saint-Etienne
6. 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“ (1979)
7. Owen A.B. (1992). "Orthogonal arrays for computer experiments, integration and visualiazation". *Statistica Sinica* 2,439-452.
8. Thiémond.E, (2000). Sur le calcul et la majoration de la discrèpance a l’origine . thèse No 2259, Département de Mathématiques , école polytechniques fédérale de Lausanne
9. D. Daley et V. Jones., “ An introduction to the théory of point processes. Springer “, New York, 1988.
10. N. M. Van Lieshout., “ Markov Point Processes and their Applications “. Imperial College Press, London, 2000
11. D. Stoyan, W.S. Kendall et J. Mecke (_editors)., *Stochastic geometry and its applications*. Wiley, Chis tester, second edition, 1995.
12. B. D. Ripley et F. P. Kelly., “ Markov point processes. *Journal of the London Mathematical Socie-ty* “, 15(188-192), 1977.

13. J. Møller et R. Waagepetersen., "Statistical Inference and Simulation for Spatial Point Processes ". Chapman and Hall/CRC, Boca Raton, 2004.
14. J. Neyman et E.L. Scott., " Statistical Approach to Problems of Cosmology. Journal of the Royal Statistical Society", series B, 20 :1–43, 1958.
15. J. Møller, A.R. Syversveen, et R.P. Waagepetersen. Log Gaussian Cox processes. Scandinavian Journal of Statistics, 25 :451–482, 1998.
16. Metropolis, N. (1987) The Beginning of the Monte Carlo Method, rapport technique no LA-UR-88-9067, Los Alamos National Laboratory.
17. Warnock, T.T., "Computational investigations of low-discrepancy point sets II", In: Niederreiter H., Shiue, P.J.S. (eds.) Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing. Lecture Notes in Statistics, vol. 106, Springer, New York, (1995).
18. Chib, S. and Greenberg, E., "Markov chain Monte Carlo simulation methods in econometrics", Econ. Theory, 12, (1996), 409–431.
19. Meyn, S.P. and Tweedie, R.L., "Markov chains and stochastic stability", Springer New York, (1993).
20. Dobrushin, R.L., "Central limite théorème for no stationary Markov chains", Th.Proba. Appl., 1, (1956), .
21. Winkler, G., "Image Analysis Random fields and Dynamic Monte Carlo Methods", Springer, Berlin, (1995).
22. . Hastings, W.K., "Monte Carlo sampling methods using markov chains and their applications", Biometrika , (1970).
23. Metropolis, N., Rosenbluth A. W., Teller A. H. and Teller E., "Equation of state calculations by fast computing machines" Journal of Chemical Physics, (1953).
24. Jaulin, F., "Processus ponctuels markoviens ", Juin 2008
25. Geyer, C.J. and Møller, J., "Simulation and likelihood inference for spatial point processes", Scandinavian Journal of Statistics, Series B, 21 :359–373, (1994).
26. Tierney, L., "Markov chains for exploring posterior distributions", Ann. Statist, 22, (1994).
27. Mira, A. (2001), *On Metropolis-Hastings Algorithms with Delayed Rejection*, Metron, Vol. LIX, n. 3-4, pp. 231-241.

28. Elmoosaoui, H. , «contribution a la méthodologie de la recherche expérimentale » , thèse de doctorat, Université de Saad DAHLEB de Blida (2020).
29. Mathieu D., ‘‘Contribution de la Méthodologie de la Recherche Expérimentale à la relation structure-activité , Thèse de doctorat , Université d’Aix-Marseille (1981).
30. Johnson, M.E., Moore, L.M. and Ylvisaker, D., ‘‘Minimax and maximin distance designs’’, J. Stat. Plann. Inference 26, (1990), 131–148
31. Gunzburger, M. and Burkardt, J., ‘‘Uniformity measures for point samples in hypercubes’’https://people.sc.fsu.edu/~jburkardt/publications/gb_2004.pdf,(2004).