

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE  
SCIENTIFIQUE

Université Blida 1

Faculté Des Sciences

Département De mathématiques



Mémoire Présenté Par

**TABOUCHE Mohammed Ayoub et MOUAS Mourad**

En vue de l'obtention du Diplôme de Master en Modélisation stochastique  
Et statistique

**THEME**

**ETUDE DE L'ALGORITHME DE METROPOLIS-HASTINGS  
POUR LA SIMULATION DES PLANS D'EXPERIENCES  
NUMERIQUES**

**MEMBRES DU JURY :**

Mme	MESSAOUDI Nadia	Université Blida 1	Présidente
Mme	RABER Dalila	Université Blida1	Examinatrice
Mr	ELMOSSAOUI Hichem	Université Blida1	Promoteur

**2019/2020**

## Remerciements

Avant tout, On remercie Dieu le tout puissant, pour nous avoir donné de la santé, le courage et la volonté d'étudier et pour nous avoir permis de réaliser ce travail dans de bonnes conditions.

Nous exprimons nos profonds remerciements à notre promoteur, Mr ELMOSSAOUI Hichem, pour l'aide qu'il nous a apporté durant des années et pour sa patience, nous le remercions vivement et nous espérons que nos efforts et nos résultats ont été à la mesure de son attente.

Nos remerciements vont également à tous nos professeurs et enseignants qui par leur compétence nous ont soutenu dans la poursuite de nos études et qui ont contribué à ce couronnement.

Nos remerciements vont à Mme Messaoudi Nadia d'avoir acceptée de présider le jury et à Mme Raber Dalila, d'avoir acceptée d'examiner ce modeste travail.

Et à tous ce qui ont participés de près ou de loin à l'aboutissement de ce travail.

## **Dédicaces**

Je dédie ce modeste travail à :

Mes très chers parents :

Ma mère, qui a sacrifié pour que je réussisse. Je la remercie pour le soutien, l'amour, et que dieu la protège et la garde pour nous.

Mon père, merci pour les valeurs nobles, l'éducation et le soutien permanent venu de toi. Tu restes toujours mon exemple de courage, d'espoir et de volonté.

Mes chères sœurs

Mon agréable frère Mohammed Ayoub

Sans oublier mon prof 'Ali Derbala' que dieu ait le pitié de lui

A mes amis : Youcef, Younes, Abd elhak ...

Et toute la promo de mathématique 2019/2020

**Mourad**

## Dédicaces

Je dédie ce modeste travail

*A mes très chers parents pour m'avoir donnés le goût aux études et m'avoir  
apportés un  
grand support moral lors de la rédaction de ce mémoire et tout au long de mes  
études,  
qu'ils trouvent ici l'expression de mon profond Amour.*

*A ma sœur et ma petite nièce Ania Mélina.*

*A mes chères grands-mères, et à mes chères tantes et oncle.*

*A mes chers cousins et cousines*

*Je ne me permettrais surtout pas d'oublier mes très chers amis(es).*

*A tous les gens qui m'ont aidé et soutenu tout au long de ce projet.*

*Et à tous ceux que j'aime . . .*

***Med Ayoub***

## Résumé

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 travail, nous nous sommes intéressés à la l'algorithme de Métropolis –Hastings et la méthode de Monté Carlo par Chaine de Markov (MCMC) utilisés dans 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.

**Les mots clés :** MCMC, l'algorithme de Métropolis–Hastings, plans d'expériences, plant d'expériences numériques.

## ABSTRACT

The design of experiments method is a set of method and mode of reasoning intended for any experimenter wishing to do experimental design, in this work, we are interested in the Metropolis algorithm - Hastings and the method of Monté Carlo by Markov (MCMC) Chain used in the construction of experimental designs from deterministic simulators and in the exploratory phase ie when the dependence between the inputs and the outputs is a priori unknown.

**Key words :** MCMC, algorithm of Metropolis–Hastings, experience, designs of experiments, computer experiments desegns.

## ملخص

طريقة التصاميم التجريبية عبارة عن مجموعة من الأساليب و طرق التفكير الموجهة لأي مجرب يرغب في القيام بالتخطيط التجريبي، في هذا العمل نحن مهتمون بخوارزمية ميروبوليس هاستن وطريقة مونتي كارلو بسلسلة ماركوف التي تستخدم في بناء التصميمات التجريبية بواسطة المحاكيات القطعية وفي المرحلة الاستكشافية، أي عندما يكون الاعتماد بين المدخلات والمخرجات غير معروف مسبقاً.

**الكلمات المفتاحية :** مونتي كارلو بسلسلة ماركوف، خوارزمية ميروبوليس هاستن، التصميمات

التجريبية، التخطيط

## Table des matières

Remerciements	
Dédicaces	
Résumé	
Liste des tableaux	
Liste des figures	
Introduction générale .....	13
<b>CHAPITRE 1 : Plans numériques</b>	
1.1. Des plans classiques aux plans numériques.....	15
1.1.1. Principes fondamentaux.....	16
1.1.1 Plans factoriels complets et fractionnaires.....	16
1.1.2 Les plans d'expériences numériques.....	18
1.2 Plans stratifiés.....	18
1.2.1.1 Grilles régulières.....	20
1.2.1.2 Hypercubes latins.....	21
1.2.1.3 Tableaux orthogonaux.....	24
1.2.2 Suites à faible discrédance.....	26
1.2.2.1. Suites de Sobol.....	27
1.2.2.2. Suites de Halton.....	29
1.2.2.3. Suites de Hammersley.....	31
1.2.2.4. Suites de Faure.....	32
<b>CHAPITRE 2 : CRITERES ET PLANS OPTIMAUX</b>	
2.1. Critères d'uniformité basés sur la discrédance.....	34
2.1.1. La discrédance.....	34
2.1.2. Approches calculatoires de la discrédance en norme 2.....	37

2.1.2.1. Discrépance extrême et à l'origine.....	37
2.1.2.2. Discrépance modifiée.....	37
2.1.2.3. Discrépance centrée.....	38
2.1.2.4. Discrépance symétrique.....	39
2.1.2.5. Discrépance « enveloppée ».....	39
2.1.4. Exemples.....	40
2.2. Critères basées sur la distance.....	41
2.2.1. Mesure de recouvrement.....	42
2.2.2. Le rapport de distances.....	42
2.2.3. Critère Maximin et Minimax.....	43
2.2.4. Exemples.....	44
 <b>CHAPITRE 3 : Méthodes MCMC et l'algorithme de métropoles Hastings</b>	
3.1. Méthodes de base de simulation.....	45
3.1.1. Intégration de Monte Carlo.....	45
3.1.2. Uniformes.....	46
3.1.3. Variables aléatoires discrètes.....	47
3.1.4. Méthodes de transformations générales.....	47
3.2. Méthodes MCMC.....	48
3.2.1. Théorie sur les chaînes de Markov.....	48
3.2.2. Méthodes MCMC.....	51
3.2.3. Algorithme Metropolis-Hastings.....	51
3.2.4. Algorithme Metropolis-Hastings avec rejet retardé.....	54
3.4. Discussion.....	56

## Liste des tableaux

**Tableau 2.1.** Discrépance en norme infinie pour  $n=16, 25, 36$  et  $49$  pour les 3 types de plans suivants : aléatoires, suite de Halton, plan factoriel en dimension 2.....41

**Tableau 2.2.** Valeurs des critères de distance pour les 3 types de plans suivants : aléatoires, suite de Halton, plan factoriel où  $n=16, 25, 36$  et  $49$ .....44

Liste des figures

**Figure 1.1.** Plan factoriel complet à deux facteurs et 12 combinaisons.....17

**Figure 1.2.** Plan aléatoire à 30 points en dimension 2.....19

**Figure 1.3.** Grille régulière à 25 points en dimension 2.....20

**Figure 1.4.** Grille régulière randomisée à  $6^2$  points.....21

**Figure 1.5.** Un échantillonnage par hypercube latin à 5 points en dimension 2.....22

**Figure 1.6.** Un hypercube latin à 20 points en dimension 2.....23

**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  $(X_1, X_2)$ .....25

**Figure 1.8.** Un plan généré par un tableau orthogonal  $OA_1(25, 5, 5, 2)$  dont les points sont randomisés et projetés sur le sous-espace  $(X_1, X_2)$ .....26

**Figure 1.9** – Les 4 partitions possibles du carré  $[0,1]^2$  pour un  $(0, 3, 2)$ -réseau en base 2. Les couples  $(\gamma_1, \gamma_2)$  engendrant ces partitions sont :  $(3, 0)$  (haut-gauche),  $(0, 3)$  (haut-droite),  $(2, 1)$  (bas-gauche),  $(1, 2)$  (bas-droite). Chacun des 8 intervalles élémentaires contient exactement 1 point.....28

**Figure 1.10.** (De la gauche vers la droite) Les 50, 250 et 500 premiers points d’une suite de Halton en bases 2 et 3.....30

**Figure 1. 11.** 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.....30

**Figure 1. 12.** (De la gauche vers la droite) Les 50, 250 et 500 premiers points d’une suite de Hammersley en base 2.....31

**Figure 1. 13.** Les 50, 250 et 500 premiers points d’une suite de Faure en dimension 2.....32

**Figure 1. 14.** Les 80 premiers points d’une suite de Faure en dimension 6, les 100 et 200 premiers points d’une suite de Faure en dimension 8, projetés sur le carré unité défini par les deux dernières dimensions.....33

**Figure 2.1.** Les 80 premiers points d'une suite de Hammersley en dimension 2 avec un sous-ensemble J défini par  $x$  et  $x'$  pour le calcul de la discrédance extrême.....35

**Figure 2.2.** Un sous-ensemble J pour le calcul de la discrédance à l'origine.....36

**Figure 2.3.** Plan factoriel et visualisation de deux rectangles parallèles aux axes n'étant pas échantillonnés uniformément.....36

**Figure 2.4.** Un sous-ensemble J pour le calcul de la discrédance centrée.....38

**Figure 2.5.** Pour  $x = (0.7,0.75)$  , le volume total des 2 sous-ensembles J est 0.6 et la proportion de points totale est  $49/80=0.6125$ . La différence entre ces deux grandeurs est donc 0.0125.....39

**Figure 2.6.** Pour  $x = (0.8,0.35)$  et  $x' = (0.3,0.8)$  , le volume des 2 enveloppes est égal à 0.275, et la proportion de points totale est  $21/80=0.26$ . La différence obtenue entre ces deux grandeurs est donc 0.015.....40

**Figure 2.7.** De la gauche vers la droite. Un plan aléatoire, une suite de Halton, et un plan factoriel complet à 49 points en dimension.....42

### INTRODUCTION GENERALE

La simulation numérique [12] 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 [14] consiste à fixer un ou plusieurs jeux de variables d'entrée, à réaliser les calculs, puis à analyser les réponses fournies par le simulateur. Malgré les progrès toujours croissants de l'informatique, ces codes de calcul sont complexes et souvent très longs à exécuter. L'utilisateur souhaite, dans ce cas, obtenir un modèle plus simple, plus rapide et qui rende compte le mieux possible des réponses du simulateur.

La construction du modèle simplifié est fondée sur la réalisation d'un certain nombre de simulations appelées **expériences numériques**. Les expériences simulées étant coûteuses en temps calcul, il est important donc de disposer de méthodes permettant d'optimiser leur planification. Pour une famille de fonctions retenues pour construire le modèle simplifié, la qualité de représentation du modèle simplifié dépendra principalement du choix de ces expériences.

Des méthodes de planification d'expériences [10] ont été étudiées par des auteurs comme Fisher, Kiefer et Box. Leurs travaux ont donné lieu à une nouvelle branche des statistiques connue sous le nom de méthodologie des plans d'expériences. La plupart des critères de ces plans utilisent le bruit de mesure qui existe dans le cadre d'expériences réelles. Les expériences simulées ne rentrent pas dans ce contexte de planification d'expériences car les simulations que l'on considère pour créer le modèle simplifié sont déterministes. Si l'on effectue plusieurs simulations avec des jeux de variables d'entrée identiques, on obtient la même réponse. On ne peut donc pas utiliser directement la méthodologie des plans d'expériences. On peut toutefois l'adapter en considérant que ce bruit expérimental tend vers zéro.

L'objectif de ce mémoire est de parvenir à étudier divers algorithmes de Metropolis-Hastings afin d'en remarquer les forces et les faiblesses, pour peut-être y voir des améliorations possibles. Avant de se lancer dans de tels objectifs, il fallait d'abord connaître un peu de théorie sur la simulation ainsi que sur les chaînes de Markov, outils nécessaires à la compréhension et à l'étude de l'algorithme Metropolis-Hastings.

Le **premier chapitre** de ce manuscrit présente la théorie de la méthodologie des plans d'expériences.

Le **deuxième chapitre** introduit un état de l'art des plans d'expériences couramment utilisés en expérimentation numérique ainsi que de certains critères pour étudier la qualité intrinsèque de ces plans au niveau du remplissage de l'espace.

Dans le **chapitre 3** nous présentons l'algorithme de Métropolis –Hastings et la méthode de Monté Carlo par Chaîne de Markov( MCMC) et le principal algorithme de simulation des chaînes Markov Monte Carlo, l'algorithme de Métropolis-Hastings.



**CHAPITRE 1**  
**PLANS NUMERIQUES**

# CHAPITRE 1

## PLANS NUMERIQUES

### 1.1. Des plans classiques aux plans numériques

Les prémices de l'utilisation de plans d'expériences remontent probablement aux expérimentations menées par [1] afin d'étudier la capacité d'une personne à différencier ou non des poids les yeux bandés. La mise en place d'un formalisme et la conception d'une méthodologie pour la construction de plans d'expériences classiques sont introduites plus tard par [2]. Bien que Fisher ait défini cette méthodologie dans le cadre de ses applications agricoles, elle sera reprise plus tard dans le cadre d'autres applications telles que des applications industrielles et militaires.

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

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

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

$$Y = \beta X^T + \varepsilon \quad (1.1)$$

Où  $Y$  est le vecteur contenant les réponses des  $n$  expériences,  $\beta = (\beta_1, \beta_2, \dots, \beta_p)$  est un vecteur de coefficients réels,  $X$  est une matrice de taille  $n \times p$   $T$  est l'opérateur transposé et  $\varepsilon = (\varepsilon_1, \dots, \varepsilon_n)$  est un vecteur modélisant l'erreur commise lors de chaque expérience. On suppose que la matrice de covariance de  $\varepsilon$  vaut  $\text{var}(\varepsilon) = \delta^2 I_n$

Où  $I_n$  est la matrice identité de taille  $n \times n$ .

Dans le modèle linéaire (1.1) le plus simple,  $X$  s'identifie au plan d'expériences  $X$  auquel on ajoute une première ligne remplie de 1. L'estimation des coefficients  $\beta_j$ , permet alors une interprétation des effets principaux des facteurs, l'effet principal d'un facteur correspondant à l'effet de ce seul facteur sur la réponse du modèle.

Pour des modèles plus complexes cherchant par exemple à prendre en compte des interactions, les termes de la matrice  $X$  sont des combinaisons des facteurs.

### **1.1.1. Principes fondamentaux**

La caractéristique majeure des plans d'expériences classiques est la présence d'une erreur expérimentale : lors de l'expérimentation, l'obtention de la réponse peut être sujette à des erreurs de mesure ou des erreurs dues aux conditions expérimentales. Tenant compte de cette caractéristique, Fisher formalise des principes fondamentaux pour construire des plans d'expériences plus robustes.

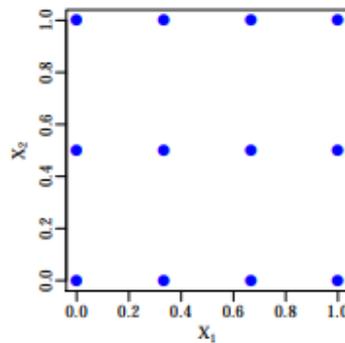
#### **1.1.1.1 Plans factoriels complets et fractionnaires**

Les plans d'expériences détaillés dans cette section sont obtenus en discrétisant indépendamment chaque facteur  $x^j, j \in \{1, \dots, k\}$ ,  $k$  nombre de facteurs, en un petit nombre de niveaux.

Un plan factoriel complet [3] est construit de façon à inclure toutes les combinaisons possibles des niveaux des facteurs, chaque combinaison contenant de l'information sur plusieurs facteurs à la fois. Ce type de plans est flexible et efficace pour étudier en même temps effets principaux et interactions entre facteurs.

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

La figure 1.1 représente les points d'un plan factoriel complet à deux facteurs avec niveaux respectifs  $(0,1/3,2/3,1)$  et  $(0,1/2,1)$ .



**Figure 1.1.** Plan factoriel complet à deux facteurs et 12 combinaisons.

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

Un plan factoriel fractionnaire est composé d'un sous ensemble des combinaisons d'un plan factoriel complet. Ces plans permettent d'estimer les effets principaux et les interactions d'ordre faible (interactions entre un petit nombre de facteurs) avec un nombre de combinaisons considérablement réduit en sacrifiant la possibilité d'estimer des interactions d'ordre élevé (interactions entre un nombre élevé de facteurs). Pour plus de détails sur ces plans, le lecteur pourra consulter les travaux de Myers et Montgomery [3].

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

### 1.1.2. Les plans d'expériences numériques

Les plans d'expériences numériques cherchent à répondre au même objectif que les plans d'expériences classiques : obtenir le plus d'informations possible sur le modèle. Cependant des différences majeures existent entre ces deux types de plans.

Les plans d'expériences numériques sont construits pour simuler une réponse d'un modèle numérique déterministe. Au contraire du modèle (1.1), les modèles déterministes sont

caractérisés par l'absence d'un terme d'erreur  $\varepsilon$ . Le modèle évalué en un point donne toujours la même réponse. L'absence du terme d'erreur est à l'origine des différences suivantes :

- Certains principes énoncés par Fisher pour la construction de plans d'expériences classiques n'ont plus lieu d'être dans le cadre de la construction d'un plan d'expériences numériques : répétition, sélection aléatoire, blocking.
- Un modèle linéaire tel que (1.1) est très souvent inadapté pour la modélisation de systèmes numériques complexes.
- Dans le cas des modèles numériques, le nombre de variables d'entrée est en principe plus élevé.
- Le domaine de variation des entrées est plus large car, sauf exception, aucune contrainte expérimentale ne permet de réduire l'ensemble des valeurs des entrées à des sous-domaines.

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

## **1.2 Plans stratifiés**

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

$$e_i = (x_{i1}, x_{i2}, \dots, x_{ik}) \quad i \in \{1, \dots, n\}$$

Où les  $x_{ij}$ , sont des variables aléatoires indépendantes uniformément distribuées sur  $[0,1]$ . Un tel échantillonnage est qualifié d'échantillonnage aléatoire, le plan d'expériences  $X$  en résultant est appelé plan aléatoire.

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

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

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

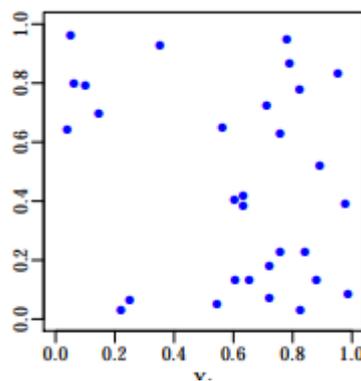
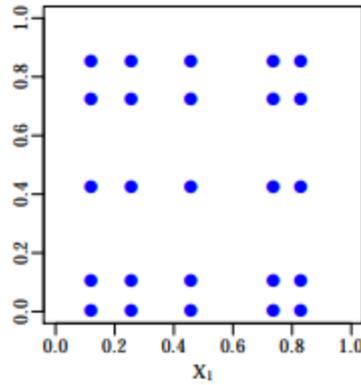


Figure 1.2. Plan aléatoire à 30 points en dimension 2

### 1.2.1.1 Grilles régulières

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



**Figure 1.3.** Grille régulière à 25 points en dimension 2

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

**Définition 1.** Soit  $X = \{e_i\}_{i=1}^n$

L'ensemble de points issus de la grille régulière construite en considérant pour chaque variable  $X^k$ , le même jeu de  $n$  valeurs  $\{1/n, 2/n, \dots, 1\}$ . Soit l'ensemble de points  $X_0 = \{Z_i\}_{i=1}^n$  construit par :

$$Z_i = (x_{i,1} - u_{i,1} / n, \dots, x_{i,k} - u_{i,d} / n) \quad ; \quad i \in \{1, \dots, n\} \quad (1.2)$$

Où les  $u_{i,j}$ , sont des variables aléatoires indépendantes uniformément distribuées sur  $[0,1]$ . Alors  $X_0$  est une grille régulière randomisée.

La figure 1.4 illustre une grille régulière randomisée  $P_0$  à  $6^2$  points construite suivant (1.2). Une grille régulière randomisée est obtenue en faisant varier chaque Point d'une grille régulière indépendamment les uns des autres à l'intérieur de leur strate respective. Ces types de plans stratifiés sont adaptés uniquement dans le cas de petites dimensions, 2 ou 3. Pour des plus grandes valeurs de  $d$ , le nombre de points créés ( $n^d$ ) croît exponentiellement et devient rapidement inadapté aux modèles requérant un temps d'évaluation important. Une alternative consiste à sélectionner un sous-ensemble des points  $Z_i$  créés suivant l'équation (1.2), de

façon à assurer une bonne répartition des valeurs prises par chaque variable  $X^k$ . Une catégorie de plans d'expériences résultant d'une telle approche correspond aux hypercubes latins

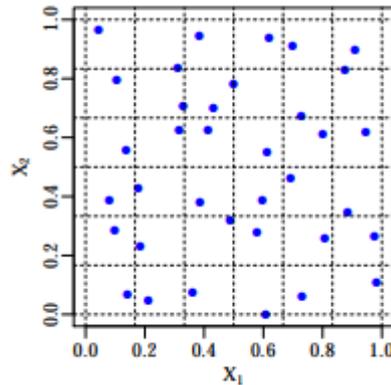


Figure 1.4. Grille régulière randomisée à  $6^2$  points

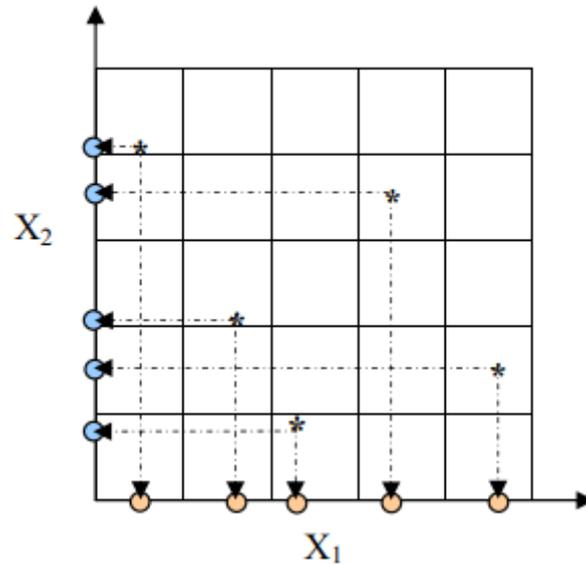
### 1.2.1.2 Hypercubes latins

En expérimentation numérique, il est courant de traiter des problèmes avec une faible taille d'échantillon (de l'ordre de la centaine) et un grand nombre de paramètres d'entrée (plusieurs dizaines). Des alternatives à la méthode de Monte Carlo ont donc été développées, non seulement pour accélérer la convergence des estimations, mais aussi pour s'assurer que les domaines de variation des paramètres d'entrée soient bien échantillonnés. En effet, placer quelques points au hasard dans un espace de grande dimension conduit le plus souvent à une mauvaise couverture de certaines dimensions.

L'une des premières méthodes développées pour contrecarrer ce problème a été celle des hypercubes latins, connue sous l'acronyme LHS pour Latin Hypercube Sampling.

Chaque axe  $[0, 1]$  du cube unité est découpé en  $n$  segments de même longueur de la façon suivante  $\left\{ \left(0, \frac{1}{n}\right), \left(\frac{1}{n}, \frac{2}{n}\right), \dots, \left(\frac{n-1}{n}, 1\right) \right\}$ . En faisant le produit cartésien de ces intervalles,

on obtient un maillage de  $n^k$  cellules de même taille. Ensuite, on choisit  $n$  cellules parmi les  $n^k$  possibles de telle manière que les cellules des marges de dimension 1 soient représentées une fois chacune. Enfin, on tire un point au hasard dans chaque cellule présélectionnée.



**Figure 1.5.** Un échantillonnage par hypercube latin à 5 points en dimension 2

**Définition 2.**

Un hypercube latin à  $n$  points sur  $[0, 1]^k$ , est défini par l'ensemble de points  $e_i$  tel que :

$$x_{ij} = \frac{\pi_j(i) + U_j^{(i)}}{n} \quad (1.3)$$

Où  $\pi_j$  est une permutation de  $1, \dots, n$  et  $U_j^{(i)} \sim U[0, 1]$  est une valeur aléatoire de distribution uniforme sur  $[0, 1]$ . Ainsi,  $(\pi_1(i), \dots, \pi_k(i))$  représente la cellule dans laquelle se trouve le point,  $e_i$  et  $(U_1^{(i)}, \dots, U_d^{(i)})$  désigne à quel endroit le point se trouve dans la cellule. L'hypercube latin obtenu correspond donc à la matrice de  $n$  lignes et  $d$  colonnes à coefficients  $X_{ij}$ .

Nous pouvons choisir de placer le point au centre des cellules pour ainsi supprimer l'aléa de la planification. Un hypercube latin, défini par la matrice  $\pi$ , est un plan très facile à construire puisque chaque colonne est une permutation de 1 à  $n$ .

Les points d'un hypercube latin ont la propriété intéressante d'être uniformément distribués sur les axes factoriels (voir Figure 1.5). Cependant, cette propriété n'assure pas l'uniformité des points sur tout le domaine expérimental. Pour  $n$  fixé, il existe  $n!$  permutations possibles pour chacune des  $k$  colonnes, et donc  $(n!)^{k-1}$  hypercubes latins éventuels qui n'assurent pas

tous l'uniformité des points dans l'espace. De façon générale, le contexte dans lequel nous sommes ne nous autorisant pas à prendre un grand nombre de points, nous aurons un nombre de mailles réduit. Il sera donc fréquent d'obtenir des plans où l'on peut observer des zones lacunaires et des agglomérats de points comme on peut l'observer sur la Figure 1.6

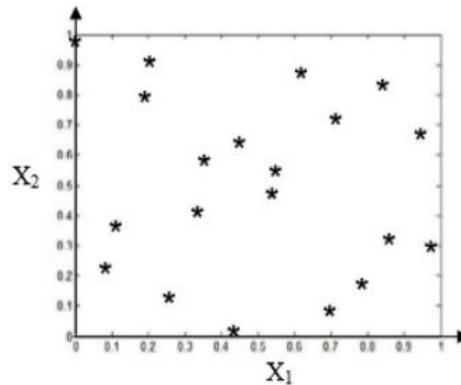


Figure 1.6. Un hypercube latin à 20 points en dimension 2

Pour pallier ces problèmes, il existe deux possibilités. La première consiste à construire les  $(n!)^{k-1}$  hypercubes latins possibles et de choisir le meilleur à l'aide d'un critère de sélection. Cependant, cette méthode n'est pas envisageable du fait du grand nombre d'hypercubes latins à générer, même si construire un hypercube latin demande peu de temps d'exécution. La deuxième possibilité, plus réaliste, consiste à générer un hypercube latin et à lui appliquer un critère afin de le rendre meilleur au sens des exigences souhaitées[20]

### 1.2.1.3 Tableaux orthogonaux

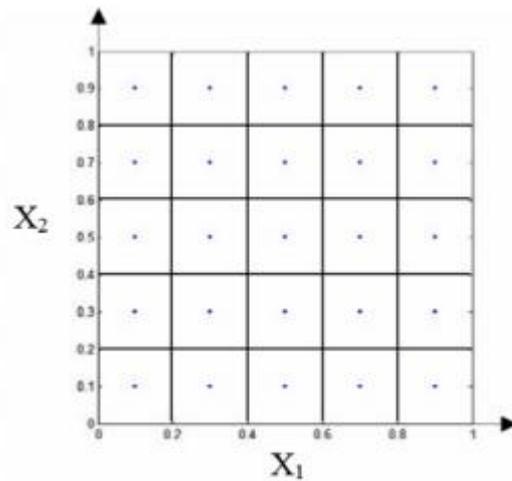
Les tableaux orthogonaux [4] sont très proches des hypercubes latins. En effet, ils ont l'avantage d'avoir les bonnes propriétés de projection en grande dimension qu'ont les hypercubes latins en dimension 1.

#### Définition 3.

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  $q \times t$  aient  $q^t$  lignes apparaissant le même nombre  $\lambda$  fois. 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  $\lambda$ -fois

On peut remarquer qu'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 4.**

Un échantillonnage par tableau orthogonal à  $n$  points sur  $[0, 1]^k$  est un ensemble de points  $e_i$  tel que

$$X_j^i = \frac{\pi_j(A_j^{(i)}) + U_j^{(i)}}{q}, \quad 1 \leq i \leq n \quad 1 \leq j \leq k \quad (1.4)$$

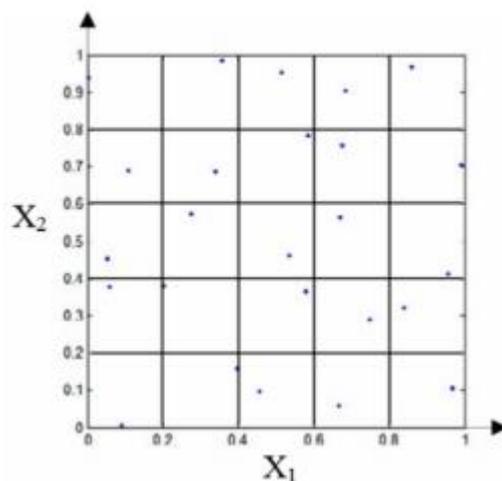
Où  $\pi_j$  est une permutation de  $0, \dots, q - 1$ ,  $A_j^{(i)}$  sont les éléments du tableau orthogonal et  $U_j^{(i)}$  est une valeur aléatoire de distribution uniforme sur  $[0, 1]$ .

Ainsi,  $(\pi_1(A_1^{(i)}), \dots, \pi_d(A_d^{(i)}))$  représente la cellule dans laquelle se trouve le point  $e_i$ , 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  $x_{ij}$ .

Concernant la génération des plans, on a une propriété analogue à celle des hypercubes latins

**Propriété 1.** 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  $(X_1, X_2)$ . Nous pouvons observer qu'il y a bien un unique point par cellule et donc que les points projetés sur le sous-espace  $(X_1, X_2)$  sont répartis uniformément



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

**Propriété 2.** Un tableau orthogonal de force  $t$  et d'index  $\lambda$  est aussi un tableau orthogonal de force  $s \leq t$  et d'index  $\lambda' = \frac{n}{q^s} \geq \lambda$

Si l'on reprend les exemples des Figure 1.7 et Figure 1.8, ces tableaux orthogonaux de force 2 et d'index  $\lambda = 1$  (1 point par cellule) sont également de force 1 et d'index  $\lambda' = 5$  (5 points par strate).

Les tableaux orthogonaux de force  $t$  permettent de généraliser aux dimensions  $t$  les bonnes propriétés de projections en dimension 1 des hypercubes latins. Ainsi, les points sont globalement répartis plus uniformément que pour les hypercubes latins.

### 1.2.2 Suites à faible discrédance

La plupart de ces suites à faible discrédance sont générées par des algorithmes déterministes afin d'obtenir des points répartis le plus uniformément possible dans le domaine expérimental. Notons que 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. Soit  $\hat{F}_n$  la fonction de répartition empirique des points  $e_1, \dots, e_n$ , on peut définir la discrédance par [5] :

$$D_n(x) = \sup_{x \in [0,1]} |\hat{F}_n - F_U(x)| \quad (1.5)$$

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

#### 1.2.2.1. Suites de Sobol

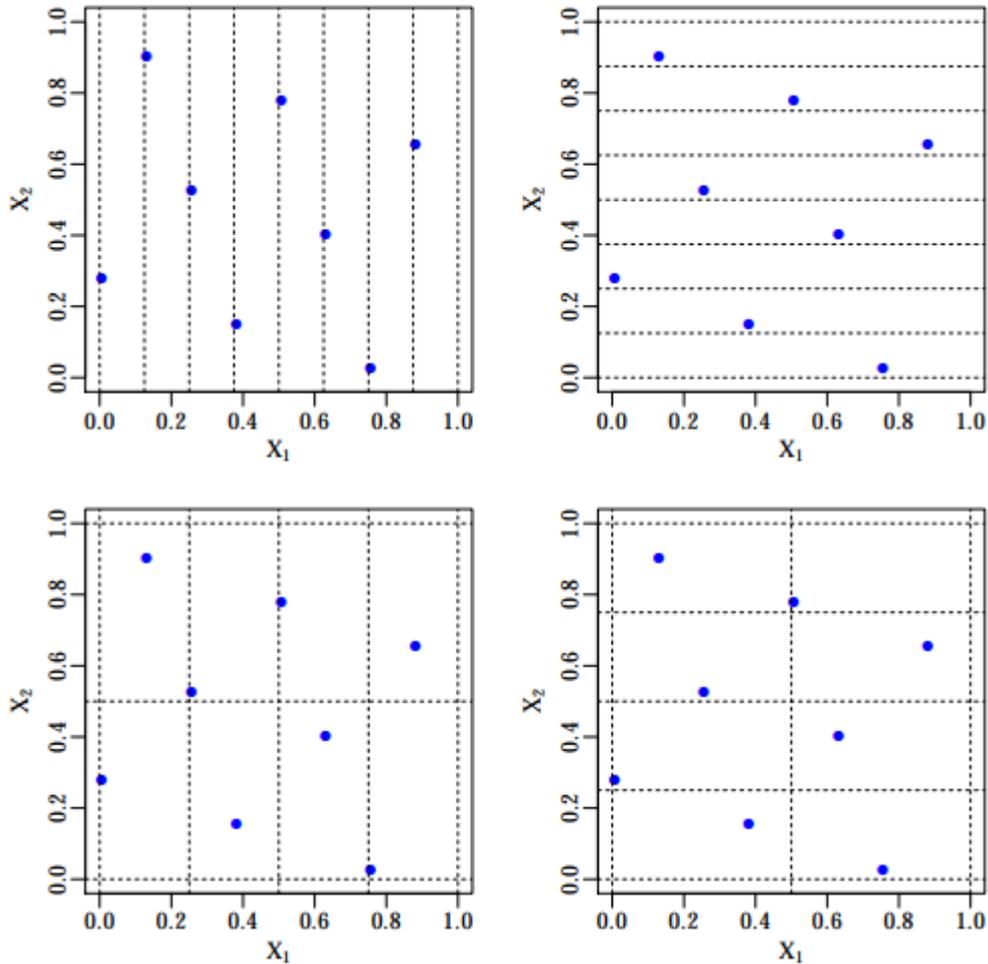
Les suites digitales en base  $b$  sont engendrées par un ensemble de matrices, appelées matrices génératrices, construites sur le corps de Galois  $GF(b) = \{0, \dots, b-1\}$ . Les suites de Sobol' sont des suites digitales en base  $k = 2$ . Pour une dimension  $k \geq 1$ , on note  $C_1, \dots, C_k$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

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

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

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

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



**Figure 1.9** – Les 4 partitions possibles du carré  $[0,1]^2$  pour un  $(0, 3, 2)$ -réseau en base 2. Les couples  $(\gamma_1, \gamma_2)$  engendrant ces partitions sont :  $(3, 0)$  (haut-gauche),  $(0, 3)$  (haut-droite),  $(2, 1)$  (bas-gauche),  $(1, 2)$  (bas-droite). Chacun des 8 intervalles élémentaires contient exactement 1 point.

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

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

Et la matrice  $C_j$  s'écrit :

$$C_j = \begin{pmatrix} 1 & v_{2,j,1} & v_{3,j,1} & \cdot & \cdot & \cdot \\ 0 & 1 & v_{3,j,2} & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Les matrices  $C_j$  sont de taille infinie puisqu'elles sont issues d'une construction récursive.

### 1.2.2.2. Suites de Halton

Les suites de Halton sont la version en dimension  $d \geq 1$  des suites de Van Der Corput qui quant à elles constituent la version unidimensionnelle de ces suites. L'idée pour générer les suites de Halton consiste à utiliser une base différente pour chaque dimension.

#### Définition

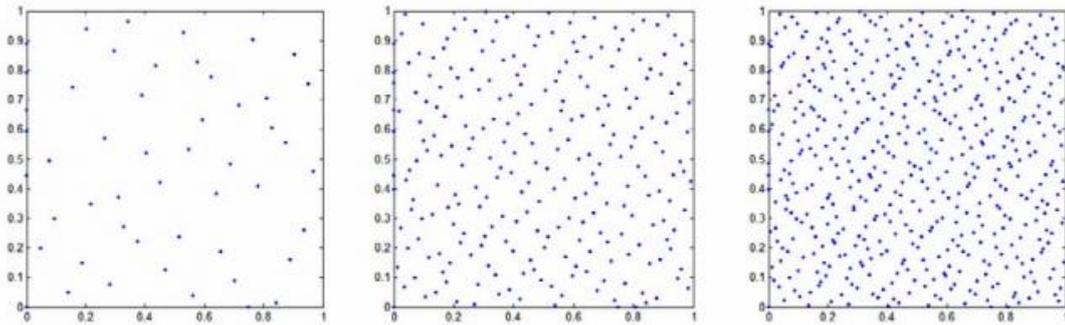
Une suite de Halton  $H_{b_1}, \dots, H_d = \{x^0, x^1, \dots, x^{n-1}\}$  en bases  $b_1, \dots, b_d$  est définie par

$$x^i = (\phi_{b_1}(i), \dots, \phi_{b_d}(i)) \in [0,1]^d$$

Où  $b_1, \dots, b_d$  sont des entiers positifs premiers entre eux deux à deux.

Afin de minimiser au mieux la discrédance, il convient de prendre pour les bases les  $d$  premiers nombres premiers. En effet, cela permet de minimiser la constante intervenant dans le terme dominant de la majoration de la discrédance de la série de Faure H. (1982) [6][20].

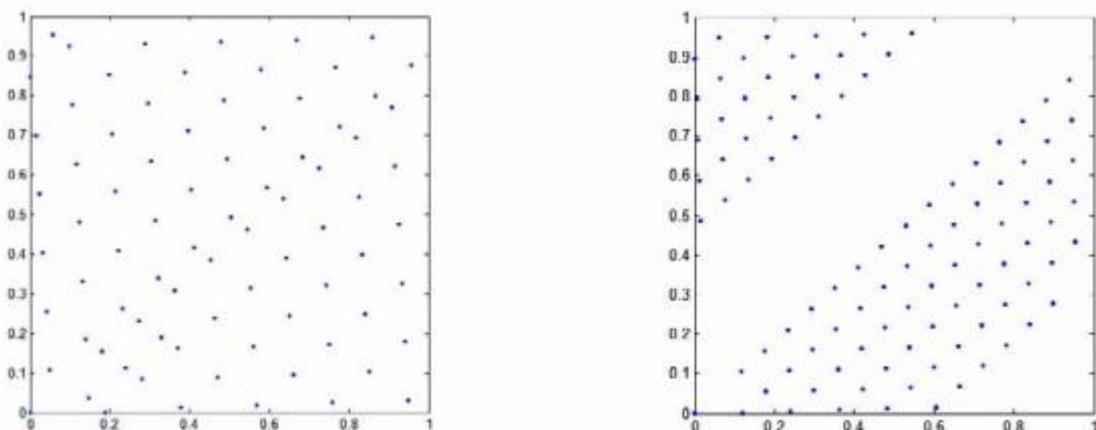
Ces suites ont l'avantage d'être faciles à implémenter et d'avoir un temps d'exécution très faible. Pour passer de  $e_i = \phi_b(i)$  à  $e_{i+1} = \phi_b(i+1)$  il suffit d'effectuer une addition, en base  $b$ , à droite de  $1/b$ .



**Figure 1.10.** (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 (Figure 1.10), 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 1.11).

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[20].



**Figure 1.11.** 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

### 1.2.2.3. Suites de Hammersley

Une suite de Hammersley en dimension  $k$  est construite à partir d'un terme dépendant du nombre de points et d'une suite de Halton en dimension  $k-1$ .

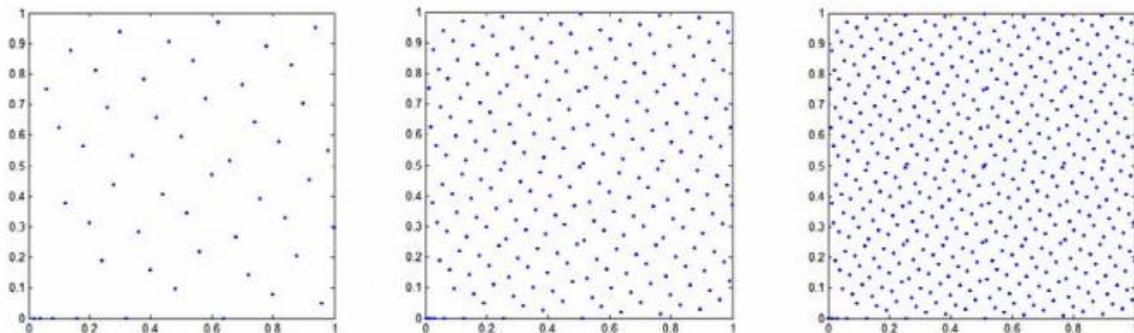
**Définition :**

Une suite de Hammersley  $H_{b_1, \dots, b_{d-1}}^n = \{e_1, \dots, e_n\}$  en bases  $b_1, \dots, b_{k-1}$  est définie par :

$$e_i = \left( \frac{i}{n}, \phi_{b_1}(i), \dots, \phi_{b_{d-1}}(i) \right) \in I^k$$

Où  $b_1, \dots, b_{k-1}$  sont des entiers positifs premiers entre eux deux à deux.

Ces suites étant construites à partir de suites de Halton, elles 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 discrèpance. 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[20].



**Figure 1. 12.** (De la gauche vers la droite) Les 50, 250 et 500 premiers points d'une suite de Hammersley en base 2

### 1.2.2.4. Suites de Faure

Les suites de Faure sont définies à partir de la fonction radicale inverse,  $\phi_b$ , et une matrice génératrice de Pascal  $C = (C_{kl})$  donnée par

$$C_{ql} = \binom{l-1}{q-1} = \begin{cases} \frac{(l-1)!}{(q-1)!(l-q)!} & siq \leq l \\ 0 & siq > l \end{cases}$$

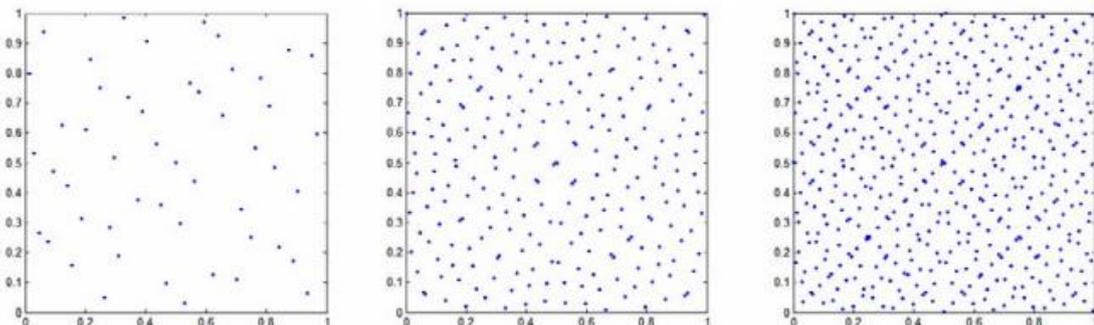
#### Définition

Soit  $b \geq d$  un nombre premier. La suite de Faure  $F = \{e_1, \dots, e_n\}$  en dimension  $d$  est définie comme suit

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

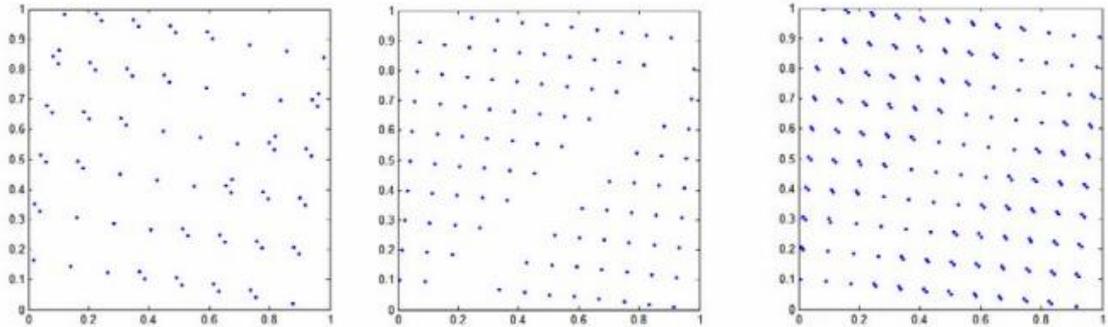
Où  $C^{j-1} = \binom{l-1}{q-1} (j-1)^{l-q} \pmod{b}$ , est la matrice génératrice de la  $j^{\text{ème}}$  dimension d'une suite de Faure en dimension  $k$ .

Il est recommandé afin d'obtenir une distribution uniforme de choisir comme base  $b$  le plus petit nombre premier supérieur ou égal à  $k$ . [20]



**Figure 1. 13.** Les 50, 250 et 500 premiers points d'une suite de Faure en dimension 2

Les suites de Faure permettent de générer une distribution uniforme localement (Figure 1.13). Cependant, les projections sur le carré unitaire des points de l'hypercube sont échantillonnées en bandes (Figure 1.14). Similairement aux suites de Sobol', nous pouvons remarquer que les nouveaux points sont placés dans le voisinage de ceux produits précédemment. Les techniques de scrambling [5] permettent également d'amoindrir ces phénomènes.[20]



**Figure 1. 14.** Les 80 premiers points d'une suite de Faure en dimension 6, les 100 et 200 premiers points d'une suite de Faure en dimension 8, projetés sur le carré unité défini par les deux dernières dimensions



**CHAPITRE 2**  
**CRITERES ET PLANS OPTIMAUX**

## CHAPITRE 2

### CRITERES ET PLANS OPTIMAUX

Dans ce chapitre nous allons référer à des critères afin de pouvoir décider si une distribution est uniforme ou non ou si elle vérifie également le bon remplissage de l'espace. Il existe pour cela des critères intrinsèques aux plans d'expériences à savoir des critères qui ne préjugent en rien de la qualité de la surface de réponse déterminée par la suite à partir de ces plans.

Aussi, nous discutons dans un premier temps les critères de discrédance qui permettent de mesurer si la distribution donnée est proche d'une distribution uniforme.

Ensuite, nous introduisons la classe des critères de distance qui comparent une distribution donnée à une grille régulière.

Enfin, nous présentons le critère d'entropie qui mesure la quantité d'information contenue dans le plan. Contrairement aux deux précédents, ce critère n'est pas intrinsèque car il dépend d'un modèle statistique. Nous verrons cependant qu'il est lié à la distance entre les points et assure ainsi une bonne distribution.

#### 2.1. Critères d'uniformité basés sur la discrédance

##### 2.1.1. La discrédance

On rappelle que la discrédance mesure l'écart entre une distribution de points donnée et une distribution de points uniforme ; en d'autres termes, elle mesure l'irrégularité de la distribution. On donne ci-dessous la définition permettant de comprendre le principe sur lequel elle est basée.

##### Définition [4]

Soit  $X$  la suite de  $n$  points  $e_1, \dots, e_n$  dans  $[0, 1]^k$  et soit  $J$  un sous-ensemble de  $[0, 1]^k$

Avec les notations précédentes on pose :  $D_n(J, X) = \frac{A(J, X)}{n} - \lambda_d(J)$

où  $A(J, X)$  est le nombre d'indices  $i$ ,  $1 \leq i \leq n$ , tel que  $x_i \in J$  et  $\lambda_d(J)$  est la mesure de Lebesgue (ou volume) de  $J$ .

La **discrédance extrême** de  $X$ , notée  $D_n(X)$  est définie par  $D_n(X) = \sup_{J \in \mathfrak{S}} |D_n(J, X)|$

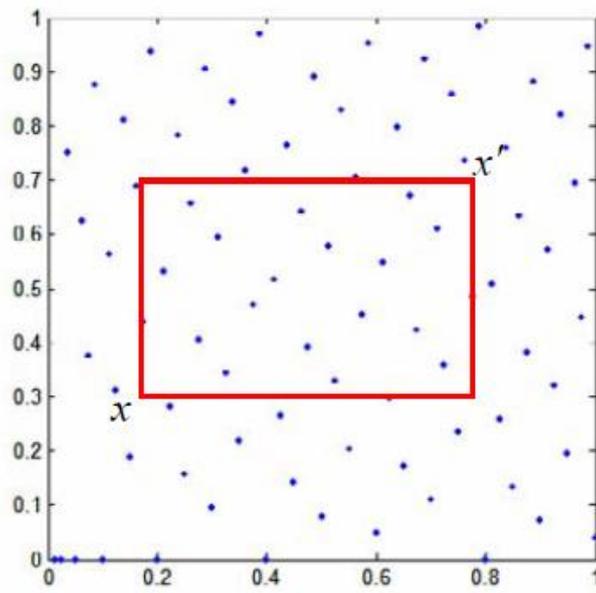
où  $\mathfrak{S}$  est l'ensemble des parties de  $[0, 1]^k$  de la forme  $\prod_{i=1}^k [a_i, b_i]$

La **discrédance à l'origine** de  $X$ , notée  $D_n^*(X)$  est définie par

$$D_n^*(X) = \sup_{J \in \mathfrak{S}^*} |D_n(J, X)|$$

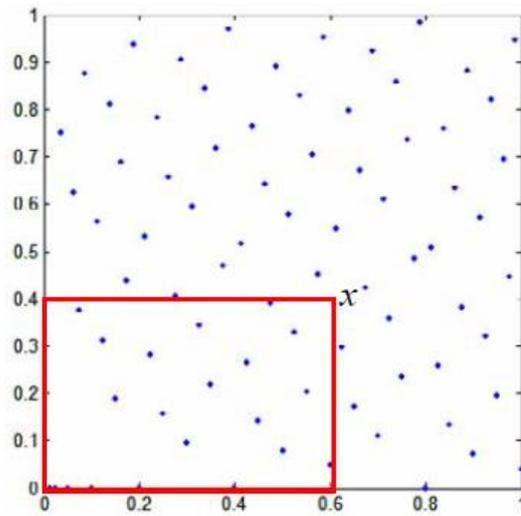
où  $\mathfrak{S}^*$  est l'ensemble des parties de  $[0,1]^k$  de la forme  $\prod_{i=1}^k [0, b_i]$

**Remarque :** Soit  $\mu_X = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$  la probabilité uniforme sur  $X$ , alors  $D_n(J, X)$  s'écrit comme la distance entre celle-ci et la mesure de Lebesgue soit  $D_n(J, X) = \mu_X(J) - \lambda_d(J)$



**Figure 2.1.** Les 80 premiers points d'une suite de Hammersley en dimension 2 avec un sous-ensemble  $J$  défini par  $x$  et  $x'$  pour le calcul de la discrédance extrême

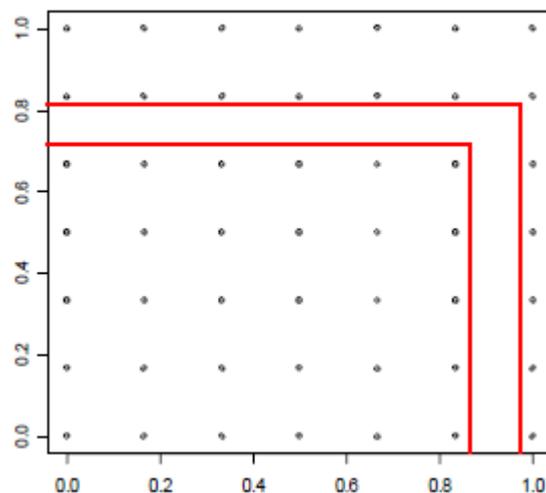
Pour  $x = (0.2, 0.3)$  et  $x' = (0.8, 0.7)$ , le volume du rectangle  $J$  est égal à 0.24, et la proportion de points dans  $J$  est  $18/80 = 0.225$ . On a donc  $D_n(J, X) = 0.015$ . En effectuant le même procédé sur plusieurs sous-ensembles  $J$  et en prenant la valeur maximale acquise, on obtient la discrédance extrême de la suite  $X$ .



**Figure 2.2.** Un sous-ensemble J pour le calcul de la discrédance à l'origine

$D_n(X)$  et  $D_n^*(X)$  sont des discrédances en norme  $L_\infty$ . Les discrédances en norme  $L_2$  définissent une nouvelle mesure du non uniformité. Elles seront données dans la section 2.1.2.

Maintenant que nous venons de voir la définition exacte de la discrédance, il devient possible d'expliquer les mauvais résultats de discrédance d'une grille. Pour en comprendre l'origine, il faut se rendre compte que pour avoir une faible discrédance, il faut échantillonner le plus uniformément possible tous les rectangles parallèles aux axes. Ainsi pour une grille, il y a des rectangles dont l'échantillonnage n'est pas bon. Par exemple, il peut y avoir des problèmes selon le placement des bords par rapport à la grille.



**Figure 2.3.** Plan factoriel et visualisation de deux rectangles parallèles aux axes n'étant pas échantillonnés uniformément

### 2.1.2. Approches calculatoires de la discrédance en norme 2

Seule la discrédance en norme  $L_2$  est facilement calculable quelle que soit la dimension. On note  $X$  la suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  Cette section présente dans un premier temps la définition des différentes discrédances en norme 2 ainsi que leurs approches calculatoires.

#### 2.1.2.1. Discrédance extrême et à l'origine

##### Définition

La discrédance  $L_2$  d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par

$$DL_{2(n)}(x)^2 = \int_{[0,1]^{2k}} (D_n(J, X))^2 da db$$

où  $J$  est l'ensemble des parties de  $[0,1]^d$  de la forme  $\prod_{i=1}^d [a_i, b_i]$

##### Définition

La discrédance  $L_2$  à l'origine d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par

$$DL_{2^*(n)}(x)^2 = \int_{[0,1]^{2k}} (D_n(J, X))^2 db$$

où  $J$  est l'ensemble des parties de  $[0,1]^k$  de la forme  $\prod_{i=1}^k [0, b_i]$

En dimension  $d$ , les calculs de  $DL_{2n}$  et  $DL_{2n}^*$  peuvent se faire par les formules explicites suivantes :[6]

$$DL_{2(n)}(x)^2 = \left(\frac{1}{12}\right)^k - \frac{2^{1-k}}{n} \sum_{i=1}^n \prod_{j=1}^k x_{ij} (1 - x_{ij}) + \frac{1}{n^2} \sum_{i=1}^n \sum_{k=1}^n \prod_{j=1}^k (1 - \max(x_{ij}, x_{kj})) \min(x_{ij}, x_{kj})$$

$$DL_{2^*(n)}(x)^2 = \left(\frac{1}{3}\right)^k - \frac{2^{1-d}}{n} \sum_{i=1}^n \prod_{j=1}^k (1 - (x_{ij})^2) + \frac{1}{n^2} \sum_{i=1}^n \sum_{q=1}^n \prod_{j=1}^k (1 - \max(x_{ij}, x_{qj}))$$

#### 2.1.2.2. Discrédance modifiée

##### Définition :

La discrédance  $L_2$  modifiée d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par

$$DL_{2^M(n)}(x)^2 = \sum_{u \neq \emptyset} \int_{[0,1]_u} (D_p(J_u, X))^2 db_u$$

avec  $[0,1]_u$  la projection de l'hypercube unité sur les composantes  $u$  étant un sous ensemble de  $\{1, \dots, k\}$ ,  $p = \text{Card}(u)$  et  $J_u$  est la projection du sous-ensemble  $J$  de la forme  $\prod_{i=1}^d [0, b_i]$

La discrédance modifiée considère les projections sur tous les sous-espaces et est définie par :

$$DL_{2M_n}(x)^2 = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n} \sum_{i=1}^n \prod_{j=1}^k (3 - (x_{ij})^2) + \frac{1}{n^2} \sum_{i=1}^n \sum_{q=1}^n \prod_{j=1}^k (2 - \max(x_{ij}, x_{qj}))$$

### 2.1.2.3. Discrédance centrée

#### Définition

La discrédance  $L_2$  centrée d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par :

$$DL_{2C(n)}(x)^2 = \sum_{u \neq \emptyset} \int_{[0,1]_u} (D_p(J_u, X))^2 dx_u$$

avec  $[0,1]_u$  la projection de l'hypercube unité sur les composantes  $u$  étant un sous ensemble de  $\{1, \dots, k\}$ ,  $p = \text{Card}(u)$  et où  $J_u$  est la projection du sous-ensemble construit à partir du point considéré  $x$  et de son sommet le plus proche.

En dimension 2, l'ensemble  $J_u$  peut être de 4 formes différentes dont un est représenté sur la Figure 2.4 . Plus généralement, en dimension  $k$ , il existe  $2^k$  cas.

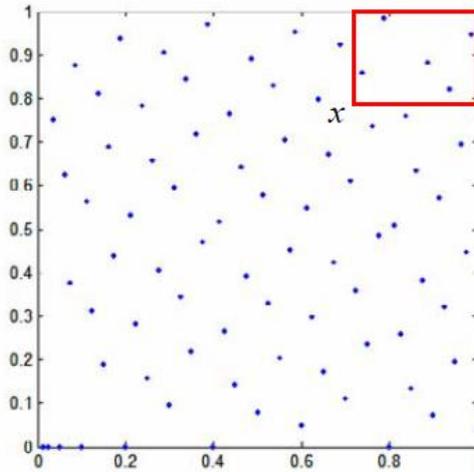


Figure 2.4. Un sous-ensemble  $J$  pour le calcul de la discrédance centrée

[7] donne une expression analytique de la discrédance centrée :

$$DL_{2C(n)}(x)^2 = \left(\frac{13}{12}\right)^k - \frac{2}{n} \sum_{i=1}^n \prod_{j=1}^k \left(1 + \frac{1}{2}|x_j^i - 0.5| - \frac{1}{2}|x_{ij} - 0.5|\right)^2 + \frac{1}{n^2} \sum_{i=1}^n \sum_{q=1}^n \prod_{j=1}^k \left(1 + \frac{1}{2}|x_{ij} - 0.5| + \frac{1}{2}|x_{qj} - 0.5| - \frac{1}{2}|x_{ij} - x_{qj}|\right)$$

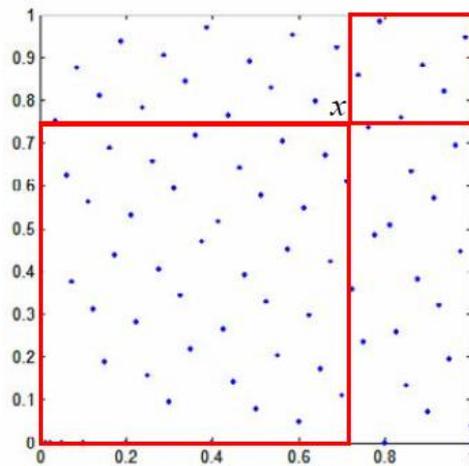
### 2.1.2.4. Discrédance symétrique

#### Définition

La discrédance  $L_2$  symétrique d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par :

$$DL_{2^S(n)}(x)^2 = \sum_{u \neq 0} \int_{[0,1]_u} (D_p(J_u, X))^2 dx_u$$

Avec  $J_u$  la projection de l'intervalle  $J$  sur le sous-espace défini par les composantes  $u$  où  $J$  est la réunion des sous-ensembles symétriques i.e. dont la somme des coordonnées des sommets sont égales.



**Figure 2.5.** Pour  $x = (0.7, 0.75)$ , le volume total des 2 sous-ensembles  $J$  est 0.6 et la proportion de points totale est  $49/80=0.6125$ . La différence entre ces deux grandeurs est donc 0.0125

On dispose également d'une formule analytique pour calculer cette discrédance :

$$DL_{2^S(n)}(x)^2 = \left(\frac{4}{3}\right)^k - \frac{2}{n} \sum_{i=1}^n \prod_{j=1}^k (1 + 2x_{ij} - 2(x_{ij})^2) + \frac{2^k}{n^2} \sum_{i=1}^n \sum_{q=1}^n \prod_{j=1}^k (1 - |x_{ij} - x_{iq}|)$$

### 2.1.2.5. Discrédance « enveloppée »

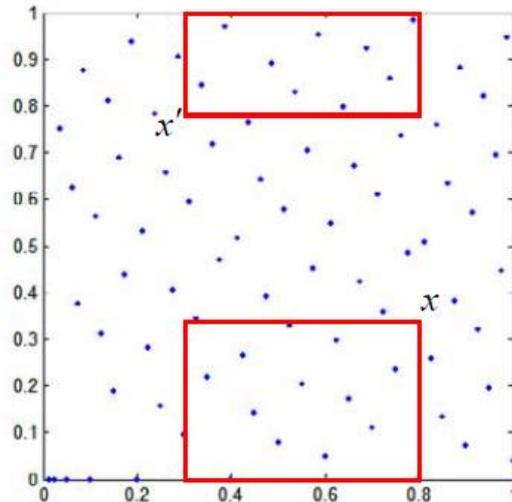
#### Définition

La discrédance  $L_2$  'enveloppée' d'une suite de  $n$  points  $e_1, \dots, e_n$  sur  $[0,1]^k$  est définie par :

$$DL_{2^S(n)}(x)^2 = \sum_{u \neq 0} \int_{[0,1]_u} (D_n(J_u, X))^2 dx_u$$

Avec  $J_u$  la projection de  $J$  sur le sous-espace défini par les composantes  $u$ , Où :

$$J = \bigotimes_{j=1}^d K(x_j, x'_j) \text{ avec } K(x_j, x'_j) = \begin{cases} [x_j, x'_j] \text{ si } x_j \leq x'_j \\ [0, x'_j] \cup [x_j, 1] \text{ sinon} \end{cases}$$



**Figure 2.6.** Pour  $x = (0.8, 0.35)$  et  $x' = (0.3, 0.8)$ , le volume des 2 enveloppes est égal à 0.275, et la proportion de points totale est  $21/80=0.26$ . La différence obtenue entre ces deux grandeurs est donc 0.015

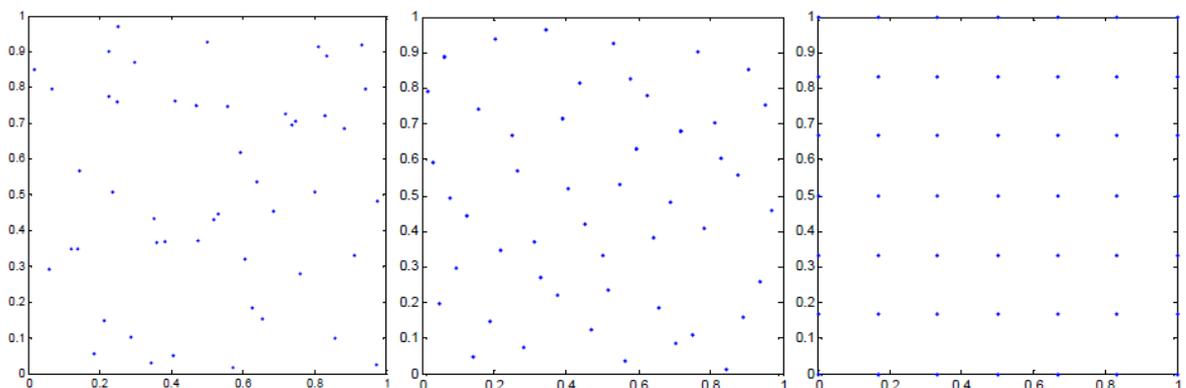
[5] introduisent la discrédance « enveloppée » comme étant la plus sensible pour détecter les zones lacunaires en projection sur les marges de dimension 1.

Ils l'utilisent pour mesurer l'uniformité des hypercubes latins réitérés. Il existe également une formule analytique pour la calculer [5] :

$$DL_{2^E(n)}(x)^2 = \left[ -\left(\frac{4}{3}\right)^k + \frac{1}{n^2} \sum_{i=1}^n \sum_{q=1}^n \prod_{j=1}^k \left( \frac{3}{2} - |x_{ij} - x_{qj}|(1 - |x_{ij} - x_{qj}|) \right) \right] \times n$$

### 2.1.4. Exemples[20]

Quelle que soit la discrédance, plus la valeur calculée pour un plan donné est faible et plus les points du plan seront distribués uniformément.



**Figure 2.7.** De la gauche vers la droite. Un plan aléatoire, une suite de Halton, et un plan factoriel complet à 49 points en dimension deux

**Tableau 2.1.** Discrédances en norme infinie pour  $n=16, 25, 36$  et  $49$  pour les 3 types de plans suivants : aléatoires, suite de Halton, plan factoriel en dimension 2

	<b>Aléatoire</b>	<b>Halton</b>	<b>Factoriel</b>
<b><math>n=16</math></b>	0.26	0.17	0.44
<b><math>n=25</math></b>	0.21	0.14	0.36
<b><math>n=36</math></b>	0.26	0.10	0.31
<b><math>n=49</math></b>	0.12	0.08	0.27

Le tableau ci-dessus montre que pour le plan factoriel, on obtient les discrédances les plus mauvaises. Ceci implique que la discrédance seule ne suffit pas à mesurer la qualité d'un plan car nous ne souhaitons pas uniquement mesurer l'uniformité des points mais aussi le remplissage de l'espace. A noter que cette remarque est valable pour n'importe quel nombre de points.

## 2.2. Critères basées sur la distance

L'idée ici est de générer des plans dont les points sont proches d'une grille régulière mais pas exactement ceux d'une grille à cause des mauvais alignements que cela peut engendrer.

On rappelle que nous recherchons des plans avec des distributions quasi-périodiques c'est-à-dire qui sont un bon compromis entre une grille régulière et une bonne uniformité (dictée par exemple avec le critère de discrédance).

On note  $dist(x^i, x^k)$  la distance euclidienne entre deux points  $x^i$  et  $x^k$  telle que :

$$dist(x^i, x^k) = \left[ \sum_{j=1}^d (x_{ij} - x_{kj})^2 \right]^{1/2}$$

### 2.2.1. Mesure de recouvrement

#### Définition

Soit  $X = \{e_1, \dots, e_n\} \subset [0,1]^k$  une suite de  $n$  points en  $d$  dimensions.

On définit la mesure de recouvrement  $\lambda$  par :

$$\lambda = \frac{1}{\gamma} \left( \frac{1}{n} \sum_{i=1}^n (\gamma_i - \bar{\gamma})^2 \right)^{1/2}$$

Avec  $\gamma_i = \min_{k \neq i} \text{dist}(e_1, \dots, e_k)$  désignant la distance minimale entre le point  $e_1$  et les autres points de la suite et  $\bar{\gamma} = \frac{1}{n} \sum_{i=1}^n \gamma_i$  la moyenne des  $\gamma_i$

Si les points sont sur une grille régulière, on a  $\gamma_1 = \dots = \gamma_n = \bar{\gamma}$  et donc  $\lambda = 0$

Ainsi, plus  $\lambda$  est faible, et plus les points sont proches de ceux d'une grille régulière. Cette expression fait apparaître explicitement le coefficient de variation de l'échantillon des  $\gamma_i$ , à savoir le rapport de l'écart-type avec la moyenne.

### 2.2.2. Le rapport de distances

#### Définition

Soit  $X = \{e_1, \dots, e_n\} \subset [0,1]^k$  une suite de  $n$  points en  $d$  dimensions.

On définit le rapport de distance par

$$R = \frac{\max_{i=1, \dots, n} \gamma_i}{\min_{i=1, \dots, n} \gamma_i} \quad \text{avec, } \gamma_i = \min_{k \neq i} \text{dist}(e_1, e_k)$$

Dans le cas où les points sont sur une grille régulière, on a  $\gamma_1 = \dots = \gamma_n$  et donc  $R = 1$ .

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

### 2.2.3. Critère Maximin et Minimax

Johnson et al. (1990) [9] ont introduit les distances maximin et minimax afin de construire des plans répondant à la question de remplissage de l'espace.

#### Définition [10]

On définit les critères suivants au sens de la distance euclidienne

$$\text{MinDist} = \min_{e_1 \in X} \min_{e_q \in X} \text{dist}(e_i, e_q)$$

$$AvgDist = \frac{1}{n} \sum_{e_i \in X} \min_{e_q \in X, k \neq i} dist(e_i, e_q)$$

Où  $X = \{e_1, \dots, e_q\}$  est le plan d'expériences à  $n$  points en  $k$  dimensions.

Maximiser le critère MinDist revient à considérer le critère de distance **maximin**. Si la valeur de MinDist est grande, alors il n'y a aucun point qui soit proche au sens de cette grandeur. Ainsi, plus les critères MinDist et AvgDist seront grands et plus cela traduira une distribution uniforme.

### Définition

On définit les critères suivants au sens de la distance euclidienne :

$$MaxDist(D, X) = \sup_{x_i \in D} \min_{x_k \in X} dist(e_i, e_q)$$

$$EDist(D, X) = \int_D \min_{x_k \in X} dist(e_i, e_q) dx^i$$

Où  $D$  est un ensemble de points du domaine expérimental et  $X = \{e_i, \dots, e_q\}$  le plan d'expériences à  $n$  points en dimension  $k$ .

L'ensemble  $D$  considéré dans la définition est généralement  $D = [0,1]^k$ . Ainsi, le supremum et l'intégrale ci-avant sont définis sur des ensembles infinis de points. Cependant, afin de calculer ces 2 expressions simplement, nous considérons un échantillonnage fini de points dans  $D$ , à savoir un plan factoriel ou un échantillonnage de ce type.

Minimiser le critère MaxDist revient à considérer le critère de distance minimax. Le principe du critère minimax est de mesurer la distance maximale entre un point du domaine expérimental n'appartenant pas au plan, qui varie, et les points du plan d'expériences. Afin de calculer les critères, on prend généralement pour les points de  $D$  une grille régulière. Aussi, en grande dimension il n'est plus envisageable de considérer ce critère sans prendre pour l'ensemble d'un plan aléatoire [11].

### 2.2.4. Exemples

Considérons à présent la valeur des critères basés sur les mesures de distance pour les différents plans de l'exemple précédent (voir Tableau 2.1 et Figure 2.7)

Tableau 2.2. Valeurs des critères de distance pour les 3 types de plans suivants : aléatoires, suite de Halton, plan factoriel où  $n=16, 25, 36$  et  $49$

$n$	Aléatoire				Halton				Factoriel			
	$\lambda$	R	MinDist	MaxDist	$\lambda$	R	MinDist	MaxDist	$\lambda$	R	MinDist	MaxDist
16	0.5	7.56	0.049	0.43	0.17	1.81	0.16	0.35	$3 \cdot 10^{-8}$	1	0.33	0.23
25	0.43	4.03	0.049	0.32	0.17	1.80	0.12	0.27	0	1	0.25	0.17
36	0.68	19.8	0.012	0.29	0.19	2.06	0.097	0.18	$3 \cdot 10^{-8}$	1	0.2	0.14
49	0.58	11.14	0.016	0.19	0.22	2.81	0.053	0.17	$2 \cdot 10^{-6}$	1	0.17	0.11

L'ensemble D considéré pour calculer le critère MaxDist est un plan factoriel à 30 niveaux ; ce qui donne dans ce cas une bonne interopérabilité à ce critère. A noter que l'ensemble D doit être formé de beaucoup plus de points que l'ensemble de points testé pour rendre compte du positionnement des points et que ces points ne peuvent pas être simplement choisis de manière aléatoire. Ce qui s'avère très difficile à assurer en grande dimension. C'est une des principales raisons qui explique que dans la suite ce critère ne sera pas étudié.



**Chapitre 3**

**CHAPITRE 3**  
**METHODES MCMC ET L'ALGORITHME DE METROPOLIS**  
**HASTINGS**

## CHAPITRE 3

# METHODES MCMC ET 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.[17]

#### 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 [12]. Elle s'énonce comme suit. Supposons que nous ayons un échantillon  $(X_1, X_2, \dots, X_N)$  généré selon la densité  $f$ . Il est alors possible d'approximer  $E_f[h(x)]$  par l'estimateur de Monte Carlo

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

Étant donné que  $\hat{h}_N$  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

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

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  $U_{[0,1]}$  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  $U_{[0,1]}$  iid (indépendantes et identiquement distribuées). Un algorithme de base utiliserait par exemple une transformation de la forme  $D(x) = \frac{ax+b}{M+1} \text{ mod } 1$ , Avec  $a, b, M \in \mathbb{N}$ .

Certains algorithmes sont souvent préprogrammés dans les logiciels et compilateurs, leur permettant ainsi d'être utilisés par une simple commande plutôt que d'avoir à en construire le code

### 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  $(X_1, X_2, \dots, X_N)$  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  $R$  (une fonction de répartition par exemple), son inverse généralisée,  $F^-$ .

Elle sera la fonction définie par

$$F^-(u) = \inf \{x : F(x) \geq u\}$$

On pourra utiliser cette inverse pour faire un premier type de transformation. En effet, si  $U \sim U_{[0,1]}$ , alors la variable aléatoire  $F^-(U)$  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  $x = F^-(u)$ . 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'on inverse 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, certain type de variables Chi

carrées, Gamma, Beta, Normales, Poisson, etc. Par exemple, l'algorithme suivant, dit Box-Muller, permet de générer des variables suivant des  $N(0,1)$  :

1. Générer  $U_1, U_2, U_{[0,1]}$  iid ;
2. Définir 
$$\begin{cases} x_1 = \sqrt{-2 \log(u_1)} \cos(2\pi u_2) \\ x_2 = \sqrt{-2 \log(u_1)} \sin(2\pi u_2) \end{cases} ;$$
3. 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 Méthodes MCMC

### 3.2.1 Théorie sur les chaînes de Markov

On appelle chaîne de Markov à temps discret toute suite de variables aléatoires  $(X_N)_{N \in \mathbb{N}}$  a valeurs dans, 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  $\pi$  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  $\pi$  est invariante pour la chaîne de Markov si :

$$\pi = \pi P$$

Cette condition est nécessaire pour obtenir la convergence de la chaîne vers  $\pi$ .

- **Réversibilité**

La chaîne est réversible pour  $\pi$  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  $\pi$ , et signifie que sous la distribution stationnaire  $\pi$  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  $\pi$ -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  $\pi$ -probable. Cette condition est clairement nécessaire pour que la chaîne converge en distribution vers  $\pi$  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  $\pi$ -irréductible et  $\pi$ -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.

### 3.2.2 Méthodes MCMC

Nous avons désormais accès à un nouveau type de méthodes de simulation, les méthodes de Monte Carlo par chaînes de Markov, ou méthodes MCMC. Ce nom désigne toute méthode qui, dans le but de simuler des variables d'une distribution  $f$ , produit une chaîne de Markov irréductible et apériodique dont la distribution stationnaire est  $f$ . Une fois cette chaîne produite, nous obtiendrons ainsi un échantillon de variables distribuées selon  $f$  et nous pourrons ainsi utiliser l'estimateur de Monte Carlo pour approximer  $E_f[h(x)]$ . La chaîne produite devra donc posséder les propriétés énoncées ci-haut.

### 3.2.3 Algorithme Metropolis-Hastings

De toutes les familles de méthodes MCMC, la plus générale est sans doute l'algorithme Metropolis-Hastings, dans le sens qu'il impose le moins de conditions sur la densité cible. Cet algorithme fut d'abord publié sous une première forme par [13], puis généralisé par [14]. À partir de la densité cible  $\pi(x)$  (possiblement en grandes dimensions), on choisit une densité instrumentale conditionnelle  $q(x, y) = q(y|x)$  à partir de laquelle il est assez facile de simuler.

Commençant avec une valeur (possiblement vectorielle)  $x_0$ , l'algorithme passe au travers des étapes suivantes à chaque itération. Sachant que la chaîne est à l'état  $x_t$  à la  $t^e$  itération,

1. Générer  $y_{t+1} \sim q(x_t, \cdot)$
2. Calculer la probabilité d'acceptation

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

3. Prendre  $X_{t+1} = \begin{cases} y_{t+1} & \text{avec probabilité } \alpha \\ x_t & \text{avec probabilité } 1-\alpha \end{cases}$

En recommençant ces étapes pour  $t$  allant de 0 à  $N$ , on construit un échantillon à partir duquel il sera possible d'estimer  $E_f[h(x)]$  par l'expression  $\hat{h}_N$ . Toutefois, pour se convaincre de la validité de cette affirmation, vérifions d'abord que la chaîne générée par l'algorithme possède les propriétés nous assurant de l'existence de sa distribution stationnaire et de la convergence de celle-ci vers la densité cible  $\pi(x)$ .

On voit rapidement que la chaîne est apériodique puisque l'évènement  $X_{t+1} = X_t$  est possible pratiquement à tout moment. En effet, chaque état peut donc être visité à deux itérations consécutives, d'où  $P_{xx}^1 > 0$ , leur période étant ainsi de 1. De plus, la chaîne est irréductible car  $q(x, y) > 0 \quad \forall x, y$  dans  $\varepsilon \times \varepsilon$  (où  $\varepsilon$  est le support de  $\pi(x)$ ), si le support de  $q$  inclut  $\varepsilon$ , ce qui devrait être le cas puisque sinon  $q$  a été mal choisie. Les états communiquent donc tous entre eux et la chaîne est ainsi composée d'une seule classe, apériodique.

Finalement, la probabilité d'acceptation a été construite pour respecter la condition de réversibilité de la chaîne. En effet, la chaîne est réversible et  $\pi$  est sa distribution stationnaire si

$$\begin{aligned} \pi(x)P(x, y)dxdy &= \pi(y)P_{yx}dxdy \\ \Leftrightarrow \pi(y)q(y, x)\alpha(y, x)dxdy &= \pi(x)q(x, y)\alpha(x, y)dxdy \\ \Leftrightarrow \frac{\alpha(x, y)}{\alpha(y, x)} &= \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)} \end{aligned}$$

En choisissant  $\alpha(x, y) = \min \left[ \frac{\pi(y)q(y, x)}{\pi(x)q(x, y)}, 1 \right]$ , la relation précédente est automatiquement

respectée. La chaîne possède donc les bonnes propriétés pour converger vers la distribution cible.

Il existe plusieurs sous-catégories d'algorithmes Metropolis-Hastings, suivant le type de distribution  $q$  choisie. D'abord, elle pourrait être indépendante de  $x_t$ , ie.  $q(x, y) = q(y)$ . La difficulté de cette version est de trouver une courbe qui épousera bien la densité cible et qui permettra une bonne exploration de l'espace. Lorsque nous cherchons une densité instrumentale aussi globale, nous sommes souvent bloqués par la complexité et par notre connaissance de la distribution cible, surtout dans les cas de grandes dimensions. C'est pourquoi il est souvent préférable (entendre plus facile) de se tourner vers une approche plus locale de l'exploration du support. Pour ce faire, on crée l'algorithme de sorte à générer une chaîne de Markov du type marche aléatoire, c'est-à-dire que la densité instrumentale  $q$  sera centrée sur la valeur courante de la chaîne et sera symétrique. La valeur proposée  $y_{t+1}$  suivra donc la forme  $x_t + \varepsilon_t$ , où  $\varepsilon_t$  est une perturbation aléatoire distribuée selon  $q$  et indépendante de  $x_t$ . Autrement dit,  $q$  sera maintenant de la forme  $q(|y - x_t|)$ . Par exemple, nous pourrions utiliser  $q \sim N(\vec{x}_t, \Sigma = I_d \cdot \delta^2)$  (où  $I_d$  représente la matrice identité en  $d$  dimensions) pour simuler  $f \sim N(0, I_d)$ .

Au-delà de l'utilisation en pratique de l'algorithme Metropolis-Hastings, on peut s'intéresser d'un œil théorique à optimiser son efficacité. Plusieurs critères s'offrent à nous pour mesurer le rendement des algorithmes. Nous avons choisi dans notre présente étude de considérer la variance de l'estimateur  $\sim N$ , ainsi qu'une valeur mesurant l'exploration du support par la chaîne créée par l'algorithme, la moyenne des distances au carré des sauts, l'ASJD (de l'anglais Average Square Jumping Distance). Celle-ci est définie par

$$ASJD = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^d (X_{i,j} - X_{i-1,j})^2$$

Où  $d$  est le nombre de dimensions de la distribution cible et  $N$  est le nombre d'itérations faites. On cherchera à la fois à minimiser la variance de l'estimateur en même temps qu'à maximiser l'ASJD (on veut que la chaîne ait exploré au maximum l'espace), ce qui s'avérera

équivalent. Un résultat théorique connu [15] stipule que, pour des distributions cibles en grandes dimensions avec composantes iid et pour des densités instrumentales symétriques, le meilleur taux d'acceptation des valeurs proposées dans l'algorithme (nombre de valeurs acceptées / nombres d'itérations) est d'environ 25% (0.234 serait la valeur théorique précise, mais le fait de viser 25% suffit dans la pratique).

La recherche de ce taux nous dirigera dans le choix des paramètres de la fonction instrumentale de notre algorithme. Par exemple, si l'on cherche à simuler  $f \sim N(0, I_{100})$  en utilisant  $q$  de la forme  $N(x_t, I_{100} \delta^2)$ , le choix de  $\delta^2$  maximisant l'efficacité de notre algorithme serait 0,0566. Plus généralement, en dimension  $d$ , le  $\delta^2$  optimal sera de la forme

$$\delta^2 = \frac{5.66}{d}$$

Pour le vérifier, il suffit de faire fonctionner l'algorithme (avec un nombre d'itérations fixe) plusieurs fois en boucle en faisant varier la valeur de  $\delta$  à chaque fois que l'algorithme recommence. On pourra ainsi trouver la valeur (ou au moins un intervalle dans lequel elle devrait se retrouver) de  $\delta$  correspondant au meilleur ASJD.

Nous avons vu, pour une certaine densité cible et après avoir choisi une famille de densité instrumentale, qu'il est possible d'optimiser l'algorithme Metropolis-Hastings dans sa forme originale en choisissant les paramètres de celui-ci. Cependant, des améliorations à l'algorithme lui-même peuvent mener à des résultats encore plus efficaces.

C'est ce qui pousse des chercheurs à proposer de nouveaux algorithmes qui le surpasseront peut-être. Le Metropolis-Hastings avec rejet retardé en est un exemple.

### **3.2.4 Algorithme Metropolis-Hastings avec rejet retardé**

Développé par [16], l'algorithme Metropolis-Hastings avec rejet retardé (en anglais, with delayed rejection) a pour but de réduire la variance de l'estimateur  $\hat{h}_N$ .

La modification proposée s'attaque au problème de la répétition d'un même état pour la chaîne lorsque la valeur proposée par l'algorithme est rejetée. Intuitivement, le fait de rester au même état trop souvent suggère que l'algorithme échoue à explorer tout l'espace et que la corrélation entre les valeurs augmente. De plus, l'information sur la valeur rejetée dans l'algorithme Metropolis-Hastings régulier ne peut être utilisée subséquentement, afin de préserver les propriétés de la chaîne de Markov. L'algorithme avec rejet retardé (ARR)

contourne cette difficulté. Le principe choisi ici pour réduire le nombre de valeurs rejetées est le suivant : lorsque la valeur proposée  $y_1$  est refusée, plutôt que d'avancer le temps en gardant la même valeur, proposons-en une seconde,  $y_2$ . Si cette valeur est à son tour rejetée, nous pourrions décider soit de garder la valeur de l'itération précédente et d'avancer le temps, soit de proposer une troisième valeur, et ainsi de suite. Les probabilités d'acceptation à chaque étape devront être ajustées (séparément) afin de préserver la propriété de réversibilité de la chaîne. L'ARR permet donc d'utiliser l'information acquise par les rejets à l'intérieur d'une même itération, offrant donc la possibilité d'un ajustement local de la distribution instrumentale.

Nous élaborerons ici la version de l'algorithme avec seulement deux étapes par itération. Soit  $\pi(x)$  la densité cible. Supposons d'abord que la chaîne soit à l'état  $X_t = x$  au temps  $t$ .

1. Générer  $y_1 \sim q_1(x, \cdot)$
2. Calculer la probabilité d'acceptation

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

3. Prendre  $X_{t+1} = y_1$  avec probabilité  $\alpha_1$ .

Si  $y_1$  est rejetée,

- (a) Générer  $y_2 \sim q_2(x, y_1, \cdot)$
- (b) Calculer la nouvelle probabilité d'acceptation

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

- (c) Prendre  $X_{t+1} = \begin{cases} y_2 & \text{avec probabilité } \alpha_2 \\ x & \text{avec probabilité } 1 - \alpha_2 \end{cases}$

Le rejet de  $y_1$  suggère que la distribution instrumentale  $q_1$  était probablement mal adaptée localement, et  $q_2$  devrait être construite en considération de cela. Cette amélioration permettra une meilleure exploration du support, et produira des valeurs d'ASJD plus grandes que le Metropolis-Hastings régulier, avec la densité instrumentale de la première étape de l'ARR correspondant à celle du Metropolis-Hastings standard.

Par contre, le temps d'exécution et de programmation pour un même nombre d'itérations sera sensiblement plus long pour l'ARR.

En cherchant à optimiser l'algorithme avec rejet retardé, on trouve, dans le cas des densités instrumentales symétriques et en grandes dimensions, que le meilleur taux d'acceptation à la première proposition est d'environ 25% (ce qui est cohérent avec les résultats obtenus pour l'algorithme régulier) et qu'il en va de même pour la seconde étape. Un exemple d'utilisation de l'algorithme serait, pour simuler une  $N(0, I_d)$ , d'utiliser  $q_1 \sim N(x_i, I_d \cdot \delta_1^2)$  et  $q_2 \sim N(x_i, I_d \cdot \delta_2^2)$ , où  $d$  est la dimension de la densité cible. Intuitivement, il paraît logique de choisir  $\delta_1 < \delta_2$ , puisque le rejet de la première valeur proposée porte à croire que la proposition était probablement trop « agressive » (trop loin de la valeur courante de la chaîne). La seconde proposition aurait ainsi avantage à être plus « conservatrice ». Dans le cas exposé, on peut démontrer théoriquement que les valeurs de variances optimales seront de la forme [14]:

$$\delta_1^2 = \frac{5.66}{d} \quad \text{et} \quad \delta_2^2 = \frac{5.66^2}{d^2}$$

Bien que cette forme de l'algorithme fonctionne assez bien en pratique (c'est-à-dire, en dimensions finies et pas trop grandes), la forme optimale de  $\delta^2$  pose cependant problème en très grandes dimensions, car les valeurs proposées à partir de  $q_2$  sont tellement proches de la valeur courante de la chaîne que la différence devient négligeable et que l'amélioration apportée par la deuxième étape face à l'algorithme régulier ne vaut peut-être plus dans ce cas le temps de calcul supplémentaire nécessaire.

### **3.4 Discussion**

Nous avons vu que l'algorithme Metropolis-Hastings avec rejet retardé, bien qu'il puisse être plus efficace en pratique que le Metropolis-Hastings régulier, pose des problèmes d'ordre théorique lorsqu'on cherche à en trouver les paramètres optimaux puis qu'il faut alors considérer ses comportements lorsque le nombre de dimensions tend vers l'infini. Toute porte à croire que de nouvelles versions pourraient être développées dans le but d'améliorer encore plus l'algorithme, tout en évitant ces difficultés théoriques et en essayant d'amoindrir le temps de calcul nécessaire. Bédard et al. [17] développent une version de l'algorithme qui fixe la

direction des propositions pour chaque itération selon la première proposition dans celle-ci. Par exemple, on pourrait choisir que si  $Y_1 = x + \delta_1 Z$  avec  $Z$  une  $N(0, I_d)$ , ie.  $Y_1 \sim N(x, I_d \cdot \delta_1^2)$ , alors  $Y_2 = x + \delta_2 Z$ , tandis que l'ARR original aurait quant à lui des propositions de la forme  $x + \delta_1 Z_1$  et  $x + \delta_2 Z_2$ . La proposition  $y_2$  serait donc dans la même direction que  $y_1$ .

L'algorithme demanderait ainsi moins de temps d'exécution que l'ARR, puisque les deux propositions reposeraient sur la même valeur  $Z$ . Les probabilités d'acceptation devraient bien sûr encore une fois être adaptées pour conserver les propriétés de la chaîne de Markov créée. Selon la même logique, on pourrait plutôt décider que la seconde proposition serait faite dans le sens opposé à la première, avec  $Y_1 = x + \delta_1 Z$  et  $Y_2 = x - \delta_2 Z$ . Les mêmes tests pourront être effectués avec ces nouvelles versions d'algorithmes pour en trouver les paramètres optimaux. Bien entendu, d'autres suggestions seront probablement faites au cours des prochaines années dans le but de rendre les méthodes MCMC toujours plus efficaces, plus rapides et plus utilisées.

## CONCLUSION GENERALE

En simulation numérique, les difficultés essentielles proviennent du coût du calcul du simulateur et de la grande dimension du problème à traiter. L'approche, actuellement privilégiée, consiste à construire un modèle d'approximation ou d'interpolation à partir d'un nombre réduit de simulations. Ces simulations sont effectuées dans des conditions fixées a priori et organisées selon un **plan d'expériences numériques**. Cette approche de modélisation de la réponse nécessite une connaissance a priori du type de dépendance entre la réponse et les paramètres et atteint ses limites lorsque la réponse étudiée est complexe. Ce travail se place dans le contexte de la phase exploratoire, à savoir lorsque cette dépendance n'est pas connue.

L'utilisation des méthodes MCMC et l'algorithme de Métropolis-Hasings 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 en termes de potentiels locaux d'énergie pour imposer des propriétés (par exemple de remplissage).

## REFERENCES BIBLIOGRAPHIQUES

1. C. S. Peirce. The doctrine of chances. Popular Science Monthly, 12 :604 615, 1878.
2. R. A. Fisher. The arrangement of field experiments. J. Ministry Agric 33 :503–513, 1926.
3. R. H. Myers and D. C. Montgomery. Response Surface Methodology : Process and Product Optimization Using Designed Experiments. Wiley, New York, NY, 1995
4. Owen A.B. (1992). Orthogonal arrays for computer experiments, integration and visualiazation. Statistica Sinica 2, 439-452.
5. Fleming J.B., Manteufel R.D. (2005). Replicated Latin Hypercube Sampling. AIAA 2005-
6. Faure H. (1982). Discrépance de suites associées à un système de numération (en dimensions), Acta Arith. 41, 33
7. Hickernell F.J. (1998). A generalized discrepancy and quadrature error bound. Mathematics of Computation, 67, 299-322. 1819.
8. Gunzburger M., Burkardt J. (2004). Uniformity measures for point samples in hypercubes. [www.csit.fsu.edu/~burkardt/pdf/ptmeas.pdf](http://www.csit.fsu.edu/~burkardt/pdf/ptmeas.pdf).
9. Johnson M.E., Moore L.M., Ylvisaker D. (1990). Minimax and maximin distance designs. J.of Statis. Planning and Inference 26, 131-148.
10. Chen V.C.P., Tsui K.L., Barton R.R, Allen J.K. (2003). A review of design and modeling incomputer experiments. Handbook of Statistics 22, 231-261.
11. Niederreiter H. (1992). Random number generation and quasi-Monte-Carlo methods. CBMS-NFS, SIAM, Philadelphia.
12. Metropolis, N. (1987) The Beginning of the Monte Carlo Method, rapport technique no LA-UR-88-9067, Los Alamos Nationale Laboratory.
13. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., e Teller, E (1953) Equations of state calculations by fast computing machines, Journal of Chemical Physics, Vol. 21, pp. 1087-1092.
14. Hastings, W. K. (1970) Monte Carlo sampling methods using Markov chains and their applications, Biometrika, Vol. 57, pp. 97-109.

15. Roberts, G. O., Gelman, A. et Gilks, W. R. (1997), Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms, *Ann. Appl. Probab.* Vol. 7, pp. 110-120.
16. Mira, A. (2001), On Metropolis-Hastings Algorithms with Delayed Rejection, *Metron*, Vol. LIX, n. 3-4, pp. 231-241.
17. Bédard, M., Douc, R., Fort, G. et Moulines, E. (2010) Scaling Analysis of Delayed Rejection MCMC Methods, rapport technique.
18. Robert, C. P. et Casella, G. (2004), *Monte Carlo Statistical Methods*, 2e édition, Springer, New York.
19. Ross, S. M. (2003), *Introduction to Probability Models*, 8e edition, Academic Press, San Diego.
20. Franco, J., “Planification d’Expériences Numériques en Phase Exploratoire pour des Codes de Calculs Simulant des Phénomènes Complexes”, Doctoral thesis, l’Ecole Nationale Supérieure des Mines de Saint Etienne, (2008).