الجمهورية الجزائرية الديمقراطية الشعبية

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

SAAD DAHLAB UNIVERSITY, BLIDA 1

FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS

DOCTORAL THESIS

Specialty: Probability and Statistics

TITLE

CONTRIBUTION TO THE METHOD OF EXPERIMENTAL DESIGN

By:

Ahmed AIT AMEUR

Supervised by

Mr. TALBI M. E.	MCA, U. Blida 1	President
Ms. BEDOUHENE F.	Professor, U.M.M.T.O	Examiner
Mr. HAMAZ A.	Professor, U.M.M.T.O	Examiner
Mr. ELMOSAOUI H.	MCA, U. Blida 1	Advisor
Ms. OUKID N.	Professor, U. Blida 1	Co-Advisor

July 3, 2025

ملخص

لقد أدت التطورات الحديثة في النمذجة، جنبا إلى جنب مع الزيادة الكبيرة في قوة الحوسبة، إلى تطوير محاكيات قادرة على تكرار الظواهر الفيزيائية بدقة لا مثيل لها. ومع ذلك، فإن تعقيد هذه المحاكيات وتكلفتها الزمنية تجعل استخدامها المباشر غالباًا غير عملي. لتجاوز هذه التحديات، من الشائع استخدام تصميمات التجارب الرقمية لإنشاء وظائف بديلة أبسط باستخدام أساليب التقريب أو الاستيفاء.

تركز هذه الدراسة على إنشاء تصميمات تجارب العددية تعتمد على العمليات العشوائية. على الرغم من أن الأساليب التقليدية قد تم دراستها على نطاق واسع، إلا أن لديها قيودًا في سياق المحاكيات الرقمية حيث يأتي الخطأ بشكل رئيسي من النموذج. لذلك، هناك حاجة إلى خطط مناسبة لتحسين تغطية المجال التجريبي واكتشاف أي عدم انتظام محتمل.

نقترح في هذه الاطروحة تصميمات تجارب عددية جديدة باستخدام العمليات النقطية المحددة وعمليات النقاط التفاعلية بالمساحة. تتضمن هذه الأساليب معرفة هندسية ومعلومات سابقة حول النقاط التجريبية، مما يسمح بتوزيع منتظم داخل مكعب الوحدة. تُستخدم بشكل محدد عمليات النقاط المميزة من نوع ستراوس ذات العلامتين وعمليات النقاط التفاعلية بالمساحة لإنشاء هذه الخطط. لهذا الغرض، نستخدم تقنيات مونتي كارلو باستخدام سلسلة ماركوف $MCMC$ ، بما في ذلك خوارزمية متروبوليس هاستينغز

الكلمات المفتاحية: الطريقة المنهج التجريبي، تصاميم التجارب العددية، العمليات العشوائية النقطية، العمليات النقطية المحددة، سلسلة ماركوف مونتي كارلو $MCMC$، خوارزمية ميتروبوليس هاستنجس

Abstract

Recent advancements in modeling, combined with a significant increase in computing power, have enabled the development of simulators capable of replicating physical phenomena with unmatched precision. However, the complexity and time cost of these simulators often make their direct use impractical. To overcome these challenges, it is common to use computer experiment designs to create simpler surrogate functions using approximation or interpolation methods.

This work focuses on the creation of computer experiment designs based on stochastic processes. Traditional methods, although extensively studied, have limitations in the context of computer simulations where the error primarily arises from the model. Therefore, suitable plans are needed to optimize the coverage of the experimental domain and detect potential irregularities.

We propose new computer experiment designs using marked point processes and area-interaction point processes. These approaches incorporate geometric knowledge and prior information about the experimental points, allowing for a uniform distribution within the unit hypercube. Specifically, Strauss marked point processes with two marks and area-interaction point processes are used to generate these plans. For this purpose, we employ Monte Carlo Markov Chain (MCMC) techniques, including the Metropolis-Hastings algorithm.

Keywords: Design of experiments, Computer experiments design, Point processes, Marked point processes, Monte Carlo Markov chain method (MCMC), Metropolis–Hastings algorithm.

Résumé

Les récentes avancées en modélisation, associées à l'augmentation significative de la puissance de calcul, ont permis le développement de simulateurs capables de reproduire des phénomènes physiques avec une précision inégalée. Cependant, la complexité et le coût en temps de ces simulateurs rendent leur utilisation directe souvent impraticable. Pour surmonter ces défis, il est courant de recourir à des plans d'expériences numériques afin de créer des fonctions de substitution plus simples, utilisant des méthodes d'approximation ou d'interpolation.

Dans ce travail, nous nous intéressons à la création de plans d'expériences numériques basés sur des processus stochastiques. Les méthodes traditionnelles, bien que largement étudiées, présentent des limites dans le cadre des simulations numériques où l'erreur provient principalement du modèle. Ainsi, des plans adaptés sont nécessaires pour optimiser la couverture du domaine expérimental et détecter d'éventuelles irrégularités.

Nous proposons dans cette thèse de nouveaux plans d'expériences numériques utilisant des processus ponctuels marqués et des processus ponctuels à interaction d'aire. Ces approches intègrent des connaissances géométriques et des informations a priori sur les points expérimentaux, permettant une répartition uniforme dans l'hypercube unité. Les processus marqués de Strauss à deux marques et les processus ponctuels à interaction d'aire sont spécifiquement utilisés pour générer ces plans. Pour cela, nous employons les techniques de simulation de Monte Carlo par chaînes de Markov (MCMC), notamment l'algorithme de Métropolis-Hastings.

Mots Clée : Plans d'expériences, Plans d'expériences numériques, Processus Ponctuels, Processus Ponctuels Marqués, Monté Carlo par Chaine de Markov (MCMC), Algorithme de Metropolis–Hastings.

To my dearly beloved mother, for her sacrifice, I wish you a long life filled with precious moments. To the memory of my father, whose life lessons have guided me here. To all those dear to me, your support has been the pillar of this academic journey.

Acknowledgments

At the end of this doctoral journey, it is my duty, filled with gratitude, to thank all those who, directly or indirectly, contributed to completing this work.

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Elmossaoui Hichem, Associate Professor at the Saad Dahlab University of Blida 1, for his unwavering support, patience, and guidance throughout this thesis. His scientific rigor, insightful advice, and constant commitment have been sources of inspiration that allowed this work to take shape. Thank you for guiding me with kindness and perseverance and believing in me throughout this journey.

I also extend my thanks to my co-supervisor, Ms. Oukid Nadia, Professor at Saad Dahlab University of Blida 1, for her valuable advice and availability. Her critical vision and encouragement helped me advance my thoughts and enrich the quality of this work. Her expertise and support were essential to the completion of this thesis.

I sincerely thank Dr. Talbi Mouhamed Elamine, Associate Professor at the University of Saad Dahlab Blida 1, for the honor of presiding over this jury. My gratitude also goes to the examiners, Professor Bedouhene Fazia and Professor Hamaz Abdelghani, from the University of Mouloud Mammeri in Tizi Ouzou, for accepting to evaluate this work and for the interest they have shown in it.

I cannot continue without warmly thanking Ms. Lamhane Kahina, former director of Atbata-Timizart High School, for her trust and support. Her leadership and commitment to academic development have been an inspiration throughout my journey.

I would like to express my sincere thanks to the entire teaching and administrative team of Saad Dahlab University of Blida 1, particularly to the Faculty of Sciences and the Department of Mathematics. Your support, guidance, and involvement greatly contributed to the success of this work. I would also like to express my gratitude to the researchers of the LAMDA-RO laboratory for their valuable exchanges, collaboration, and availability. Your expertise and encouragement were a precious source of inspiration throughout this journey.

My deep respect and gratitude also go to all the teaching staff at Atbata-Timizart High School, with whom I had the pleasure of working. Your advice, remarks, and encouragement were invaluable. Thank you to each one of you for your collaboration, availability, and spirit of sharing.

To all the university professors who have shaped my academic journey, I also express my sincere thanks. You have awakened my intellectual curiosity and passion for research, which led me to this stage of my journey. Thank you for everything you have taught me and for generously sharing your knowledge.

I would also like to thank my family, who has been a pillar throughout this journey. To my dear mother, Malika, who has always supported me, encouraged me, and instilled in me the values of perseverance and hard work. Your unconditional love and sacrifices have allowed me to achieve my goals. To my sisters, Fatiha and Lilia, thank you for your presence, your encouragement, and your moral support during difficult times.

Finally, I wish to thank all the people who contributed, in one way or another, to the completion of this thesis. Whether through a word of encouragement, a thoughtful review, or a simple smile, each of your actions mattered. This thesis is also the result of all these encounters and exchanges.

To all of you, I express my deepest gratitude.

Table of Contents

List of Figures

List of Tables

List of Algorithms

General Introduction

Recent advances in modeling, combined with the increasing computational power of modern computers, have led to the development of simulators with unprecedented complexity and accuracy. These sophisticated tools now enable highly precise modeling of a wide range of physical phenomena and complex systems, including climate dynamics, biological processes, and simulations of economic and industrial systems. By using these simulators, it is possible to achieve more reliable predictions and gain a deeper understanding of the underlying mechanisms of many natural and artificial phenomena. However, this growing sophistication comes with significant drawbacks. The computational and data management requirements increase proportionally, making these tools resource-intensive and challenging to deploy in practical contexts. Modern simulators demand high-performance computing infrastructures to carry out complex operations, presenting substantial challenges related to performance, memory management, and computation time. To address these challenges, a commonly adopted approach is to use surrogate models, also known as reduced or approximate models, to replace the original simulators when rapid solutions are required. These approximate models are typically constructed using approximation or interpolation methods based on meticulously designed computer experiment designs. Such designs, often derived from advanced techniques like orthogonal array methods or adaptive sampling strategies, capture the essential behavior of the simulated system while significantly simplifying computations. The goal is to maintain acceptable accuracy while substantially reducing the required resources. By adopting these strategies, not only are computational and time costs reduced, but data management also becomes more efficient, enabling broader and more accessible use of these advanced modeling tools.

Experiment designs have been extensively explored by researchers such as Fisher [1], Kiefer [2], and Box [3], among others. Fisher, for instance, developed methods to optimize the placement of experimental points to obtain reliable estimates of model parameters. Classical designs often prioritize placing points at the boundaries of the study domain to better capture random variations and improve the reliability of results in the presence of measurement errors. However, in the context of computer experiments, errors are often more related to model specification than to the experimentation itself. As a result, repeating classical experiments becomes less relevant, since the outcomes do not vary with experimental conditions.

The literature offers a wide diversity of experiment designs due to the lack of a universally optimal design that can simultaneously satisfy all optimality criteria. Each design has specific advantages for certain criteria while presenting limitations for others. For instance, some designs may provide better coverage of the experimental domain, while others may be more effective in detecting irregularities. Therefore, it is essential to select an experiment design based on the specific objectives of each study, considering the problem's characteristics and precision requirements.

In this context, our primary objective is to develop new computer experiment designs based on stochastic process theory. Specifically, we focus on applying marked point processes [4] and area-interaction point processes [5]. These stochastic processes offer a rigorous framework for modeling not only the spatial distribution of points but also the complex relationships between them, incorporating geometric information and prior knowledge specific to the n experimental points that constitute the computer experiment design. By employing this methodology, we achieve a refined and nuanced representation of the interactions between points, leading to improved coverage of the experimental domain. This ultimately enhances the quality and precision of computer experiments.

For the design of these experiment designs, we pay particular attention to two-mark Strauss processes [6]. These processes are used to model specific interactions between pairs of points, differentiating the types of interactions based on the characteristics of the points themselves. Simultaneously, we utilize area-interaction point processes [7], which incorporate interactions over defined regions, allowing for more precise modeling of local phenomena within the experimental domain. The goal is to ensure an optimal distribution of experimental points within the unit hypercube, providing a uniform and exhaustive coverage of the study domain. To generate these proposed designs, we implement advanced simulation techniques based on the Markov Chain Monte Carlo (MCMC) method, with a particular emphasis on the Metropolis-Hastings algorithm [8,9] combined with Voronoi tessellations [10]. These tessellations partition the space into Voronoi cells derived from a set of generating points, where each cell corresponds to the region closer to its generator than to any other. These stochastic simulation techniques are essential for efficiently exploring the space of possible configurations and identifying optimal arrangements of experimental points. They allow us to account for both the structural constraints of the problem and the specific objectives of each design, ensuring that the results are not only theoretically robust but also applicable to real-world experimental scenarios.

This document is organized to comprehensively present the various aspects of computer experiment designs and stochastic simulation. Chapter 1 introduces the fundamental principles of experimental design, explaining the basic concepts and several classical methods employed in this field. Chapter 2 delves deeply into the theory of point processes, covering essential mathematical aspects, finite point processes, Markov point processes, and fundamental concepts of random tessellations, particularly Voronoi tessellations, which are crucial for spatial modeling. Chapter 3 explores the mathematical and computational foundations of simulation, focusing on the Markov Chain Monte Carlo (MCMC) method, a powerful technique for sampling and simulating complex distributions. Chapter 4 addresses the primary objective of this research: the development of computer experiment designs based on two-marked point processes, enabling improved modeling of interactions. Chapter 5 focuses on the second objective of this thesis: designing new computer experiment designs using area-interaction point processes and developing an innovative simulation method based on the Metropolis-Hastings algorithm and Voronoi tessellations, aimed at optimizing simulation accuracy and efficiency. Finally, the work concludes with a summary and perspectives for future research. The appendix includes `Python` programs developed to produce the numerical illustrations presented in Chapters 4 and 5.

Chapter 1

Overview of Experimental Designs

This chapter synthesizes and summarizes the various assumptions underlying the use of experimental design methods. These methods are invaluable tools for any experimenter, whether conducting scientific research or industrial studies. They find application across a wide range of disciplines, whenever the goal is to determine the relationship between a variable of interest y and explanatory variables x_i that may influence it. To achieve this objective, it is essential to adhere to strict mathematical rules and adopt a rigorous approach. Such rigor ensures not only the validity of the results obtained but also their reproducibility, a fundamental aspect of any scientific process.

1.1 Historical Background

The method of experimental designs has both ancient and modern origins [11]. In the Middle Ages, Nicolas Oresme (1325–1382) mentioned this method in his writings. Later, Francis Bacon (1561–1626) [12], who inspired Descartes and Leibniz, emerged as a pioneer of the experimental method. A significant revival of this approach occurred through the work of Ronald Fisher [1], who, in 1919, was recruited by an agricultural research center near London. Faced with the impossibility of conducting all the experiments needed to improve agricultural yields, Fisher proposed experimental configurations based on rigorous statistical models, such as Latin squares.

Many statisticians, including Yates, Youden, Cochran, Plackett, and Burman, followed Fisher's contributions to promote and develop experimental planning techniques across various fields beyond agronomy. In the 1950s, the work of Box [3] and his collaborators, notably building on Yates' contributions, led to specific methods for constructing two-level fractional designs. However, it was the work of Taguchi and Masuyama that enabled a broader dissemination of experimental designs. In 1959 and 1961, they published tables for constructing orthogonal experimental designs suited to most industrial problems [13].

Many contemporary researchers continue to develop this branch of statistics in various directions: adapting experimental designs for mixtures [14], introducing block effects [15], using nonlinear models [16], incorporating neighborhood effects, and extending experimental designs to computer experiments.

1.2 Significance of Experimental Design Methods

To study a phenomenon, the experimenter focuses on a specific metric, such as the wheat yield of a field, the production cost of a chemical product, or the wear on an engine part, among others. This metric depends on multiple variables. Studying the phenomenon involves measuring this metric as a function of the various values taken by these variables.

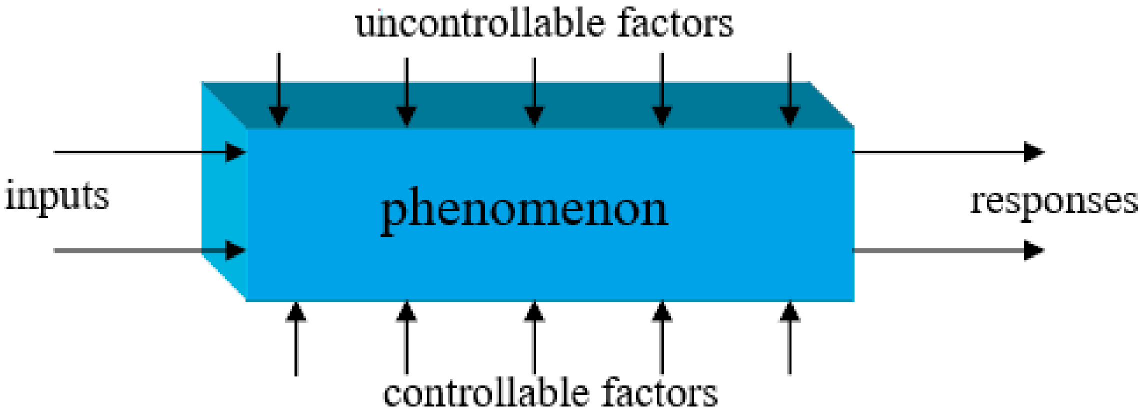

Figure 1.1: System environment.

When trying to understand the dependence of an output variable y of a process (Figure 1.1) or a property, several questions arise:

- What are the most influential factors?
- Are there interactions between the factors (correlations)?

- Can the process (or property) be linearized as a function of these factors, and is the resulting model predictive?
- How can the number of measurement points for the process (or property) be minimized while maximizing the information obtained?
- Are there biases in the measurement results?

The experimental design method addresses these questions and applies to various processes and properties, ranging from simple tests to evaluating complex quality processes. Experimental designs allow for studying numerous factors while keeping the number of trials manageable. One of their primary applications is identifying influential factors. Understanding this method relies on two essential concepts: the experimental space and the mathematical modeling of the metrics studied [17].

1.3 What is an Experimental Design?

An experimental design (commonly called 'Design of Experiments' or DOE) is an ordered sequence of trials in an experiment, each designed to acquire new knowledge by controlling one or more input parameters to achieve results that validate a model while minimizing resource use (i.e., reducing the number of trials).

Many processes and properties depend on a large number of external parameters (factors) without pre-existing mathematical models.

1.3.1 Basic Vocabulary of Experimental Designs

The methodology of experimental designs uses specific terminology in experimental research. While these terms are standard, their meanings may vary slightly across different statistical fields. To clarify our presentation, let us review some key terms.

Objectives

The primary goal of this method can be summarized by the motto: **"obtain maximum information with a minimum number of experiments."** This means producing the highest quality outcomes at the lowest possible cost, a universal objective for all manufacturers.

Factors

Factors are the variables under study that are presumed to influence the system. The value assigned to a factor for a given trial is called its "Level." A factor can be:

- **A controllable factor:** A manageable, adjustable, and modifiable variable.
- **An uncontrollable factor:** A variable not considered for the study, either because it is deemed uninfluential and left at its usual value, or because it is an unknown factor affecting the experiment.
- **A quantitative factor:** A measurable numerical quantity, such as speed, temperature, or intensity.
- **A qualitative factor:** A factor not directly quantifiable, identifiable by its different levels, such as a brand, process, method, or supplier.

When studying the influence of a factor, its variation is generally constrained between two bounds (lower bound: low level; upper bound: high level).

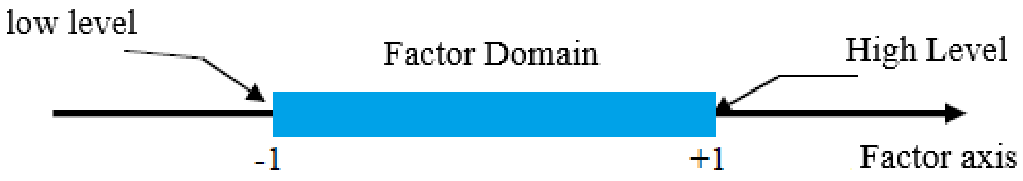

Figure 1.2: Range of factor variation.

Factor coding allows standardizing all factors to the same interval, removing their units, enabling their comparison, and simplifying subsequent mathematical analysis. Coding is performed so that the values ±1 are consistently associated with the high and low levels of operational values. This new variable is called the **centered and reduced variable**. The transformation between the original variables z and the centered and reduced variables x, and vice versa, is given by the following formula:

$$x = \frac{z - z_0}{\text{step}}$$

where:

$$z_0 = \frac{\text{high level} + \text{low level}}{2} \quad \text{and} \quad \text{step} = \frac{\text{high level} - \text{low level}}{2}$$

The advantage of centered and reduced variables lies in their ability to present experimental designs uniformly, independent of study domains and factor units. This approach lends great generality to the theory of experimental designs [18].

Experimental Domain

Consider a set of p quantitative factors used to best explain a phenomenon. The i^{th} factor (for $1 \leq i \leq p$) typically takes values within an interval of the form $[a_i, b_i]$, where a_i is the low level and b_i is the high level.

For p factors, an experiment is entirely defined by a vector in $\mathbb{R}^p$ specifying the levels of all factors. The experimental domain is any subset of $\mathbb{R}^p$ where experiments can be performed. To define such a domain, the ranges of variation for different factors are combined (see Figure 1.3).

$$[a_1, b_1] \times [a_2, b_2] \times \ldots \times [a_p, b_p]$$

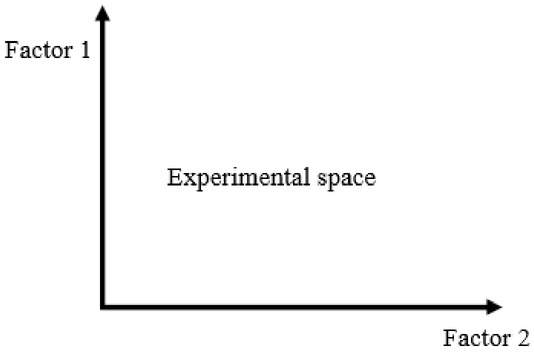

Figure 1.3: Experimental space.

The level x_1 of factor 1 and the level x_2 of factor 2 can be viewed as the coordinates of a point in the experimental space (Figure 1.4).

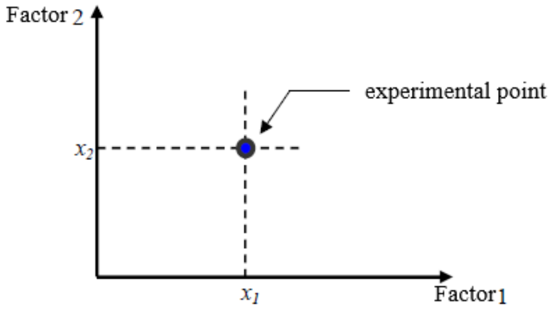

Figure 1.4: Factor levels defining experimental points in the experimental space.

An experiment is thus represented as a point within this coordinate system. An experimental design is represented as a set of such experimental points.

Response

The response is the observed value for each conducted experiment. It is assumed to always be numerical, with only one response observed at a time.

Study Domain and Response Surface

The grouping of factor domains defines what is known as the "study domain." By considering the definition of the p factors and their respective ranges of variation, it is natural to define a p-dimensional space where each point represents a configuration of the p factors. This space is referred to as the study domain or research space. Experimental points can lie either within or on the boundaries of the domain (see Figure 1.5).

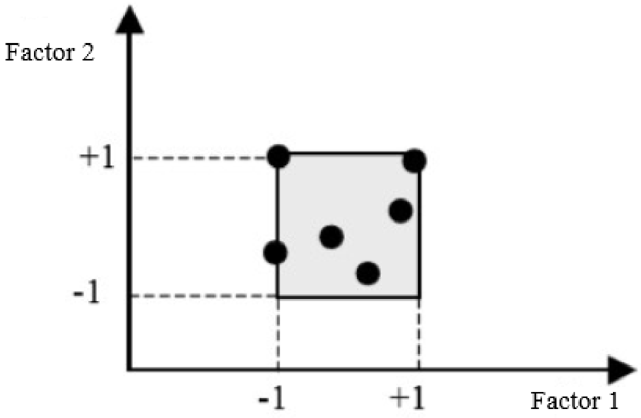

Figure 1.5: Study domain for two factors.

Each point in the study domain corresponds to a response. The collection of all points in the study domain corresponds to a set of responses, which lie on a surface called the response surface. There are two types of response surfaces:

- **Real response surface**: The real response surface of the process is the set of values the response takes.
- **Theoretical response surface**: In cases where the variables are continuous, a theoretical response surface can be calculated. In practice, this surface is constructed from a few experimental points chosen by the experimenter. Generally, the fundamental problem of experimental designs is to find a polynomial model that provides a better approximation of the real response surface (see Figure 1.6).

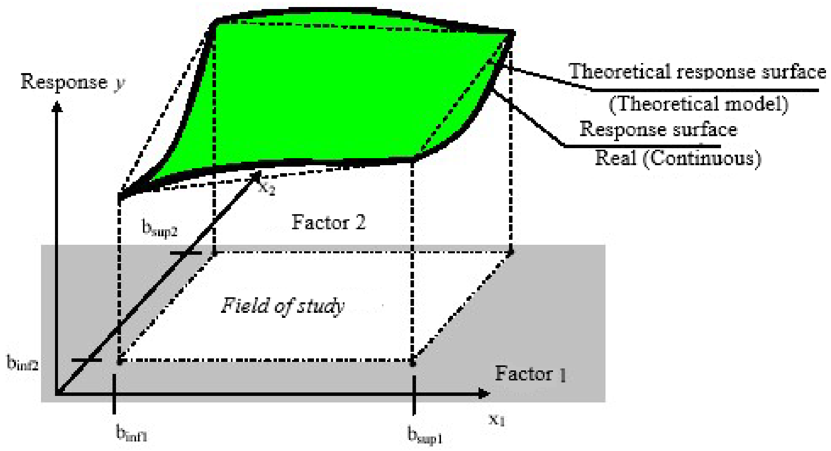

Figure 1.6: Response surface.

1.3.2 Experimental Designs

Each point in the study domain represents possible operating conditions, meaning an experiment that the operator can perform.

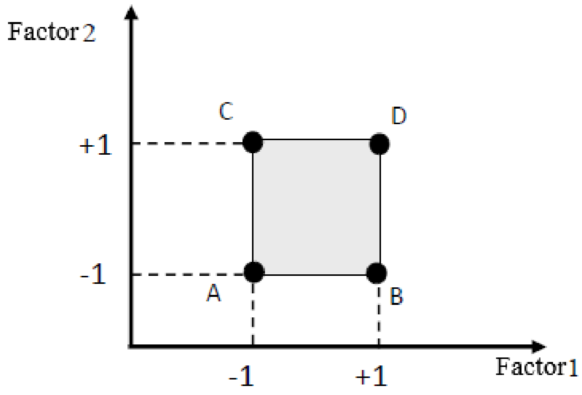

Figure 1.7: Experimental design theory shows that the best locations are the vertices A, B, C, and D of the study domain.

The selection of the number and placement of experimental points is the fundamental problem of experimental designs. Sets of experimental points meeting specific properties are referred to as classical experimental designs. These designs are well-known and widely published. When the experimental points are arranged differently from classical designs, they are called non-conventional designs. Their properties are typically less optimal than those of classical designs. However, these designs are encountered because it is not always possible to adhere to the requirements of classical designs [19].

1.3.3 Experiment Matrix

An experiment matrix is a mathematical object that represents the set of experiments to be conducted in a coded or standardized form. It is a table consisting of n rows, corresponding to the n experiments, and p columns, corresponding to the p variables (factors) under study. The experiment matrix (Tableau. 1.1) defines the trials shown in Figure 1.7. The ij-th element of the matrix corresponds to the level value of the j-th variable in the i-th experiment. The experiment matrix thus defines the trials to be carried out. The term “trial” is equivalent to “experimental point” [20].

Tableau 1.1: Experiment Matrix

Trial N°	Factor 1	Factor 2
1(A)	-1	-1
2(B)	+1	-1
3(C)	-1	+1
4(D)	+1	+1

1.4 Model and Mathematical Tools Used (Statistics)

1.4.1 Concept of Estimator

Consider a random phenomenon dependent on an unknown parameter $\beta \in \mathbb{R}$. After performing n experiments, which are realizations of random variables $y_1, y_2, \ldots, y_n$, an estimator of β is any random variable y such that $y = f(y_1, y_2, \ldots, y_n)$, where f is a known function (in other words, f must not depend on the unknown parameter β). The estimator of β is typically denoted as $\hat{\beta}$. Two classic properties of an estimator are:

Definition 1.1. *We say that:*

1. *An estimator of β is unbiased if and only if:* $\mathbb{E}(\hat{\beta}) = \beta$
2. *If $\hat{\beta}_1$ and $\hat{\beta}_2$ are two unbiased estimators of β, then $\hat{\beta}_1$ is **more efficient** than $\hat{\beta}_2$ if and only if* $\mathrm{Var}(\hat{\beta}_1) \leq \mathrm{Var}(\hat{\beta}_2)$

A good estimator of β is both unbiased (i.e., "centered" on the target to be reached) and as efficient as possible (i.e., the least dispersed around the target).

1.4.2 Statistical Model

Consider a random phenomenon dependent on p variables (factors in the case of experimental designs), and suppose that we seek to find the best mathematical model to represent this phenomenon. The statistical approach consists of conducting n experiments, wisely chosen in the case of experimental designs. Each of these is identified by a point $x \in \mathbb{R}$ (this is possible if the variables studied are quantitative; for qualitative variables, a subset of $\mathbb{N}^p$ is used). Let $y(x)$ denote the measured response such that:

$$y(x) = f(x) + \varepsilon(x)$$

where f is the deterministic part of the model, and ε is the error part, generally assumed to be a random variable under the following hypotheses:

$$\begin{aligned} &\text{Centered:} \quad \mathbb{E}(\varepsilon(x)) = 0 \quad \forall x, \\ &\text{Independent:} \quad \operatorname{Cov}(\varepsilon(x), \varepsilon(x')) = 0 \quad \forall x \neq x', \\ &\text{Homoscedastic:} \quad \operatorname{Var}(\varepsilon(x)) = \sigma^2 \quad \forall x. \end{aligned} \tag{1.1}$$

These three hypotheses aim to simplify the analysis of the models being studied.

The issue at hand is to estimate the unknown parameters of the model given by the function f (assuming the number of parameters to estimate is p').

It is essential to distinguish between linear models (i.e., linear with respect to the unknown coefficients) and non-linear models. Mathematically, a model is linear with respect to the parameters β_i ($i \in \{1, 2, \ldots, p'\}$) if and only if each of the partial derivatives $\frac{\partial f(x)}{\partial \beta_i}$ does not depend on β_i. For a random phenomenon to explain, typically the function f is unknown or too complex, so it is common to approximate it using a class of standard functions, such as Taylor expansions, Fourier series, etc.

1.4.3 Linear Modeling

Consider a statistical model dependent on p factors with f as a function that is linear with respect to p' unknown parameters. If n experiments are performed, identified by points $(z_i)_{i=1,\cdots,n}$ in $\mathbb{R}^p$, we have:

$$\forall i = 1, \cdots, n, Y(z_i) = f(z_i) + \varepsilon(z_i)$$

Since f is linear with respect to the unknown parameters, this model can be written in matrix form as:

$$Y = X\beta + \varepsilon$$

where:

- $Y \in \mathbb{R}^n$: vector of observations,
- $\varepsilon \in \mathbb{R}^n$: vector of errors,
- $\beta \in \mathbb{R}^{p'}$: vector of unknown model parameters,

- $X \in \mathcal{M}(n, p')$: model matrix.

The assumptions in (1.1) translate as:

$$\mathbb{E}(\varepsilon) = 0 \quad \text{and} \quad \mathbb{V}(\varepsilon) = \sigma^2 I_n \tag{1.2}$$

This implies:

$$\mathbb{E}(Y) = \mathbb{E}(X\beta + \varepsilon) = X\beta$$

and

$$\mathbb{V}(Y) = \mathbb{V}(X\beta + \varepsilon) = \sigma^2 I_n,$$

indicating that $X\beta$ is the mean response given by the model.

1.4.4 Parameter Estimation Using the Least Squares Method

Once the model is established, the task is to determine an estimator $\hat{\beta}$ of β that satisfies the conditions in Definition 1.1. The classical approach seeks $\hat{\beta}$ such that the vector of observed responses Y and the vector of predicted mean responses $\hat{Y} = X\hat{\beta}$ are as close as possible.

This leads to the least squares estimator of β, defined as follows:

Definition 1.2. *$\hat{\beta}$ is the least squares estimator of β if and only if it minimizes the function:*

$$Q(\beta) = \|Y - X\beta\|^2,$$

where $\|\cdot\|$ is the Euclidean norm in $\mathbb{R}^n$.

The least squares estimator of β minimizes Q, and the minimum value of Q is given by:

$$Q(\hat{\beta}) = \|Y - X\hat{\beta}\|^2 = \|Y - \hat{Y}\|^2 = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2.$$

This shows that $Q(\hat{\beta})$ represents the quadratic error between observed responses Y_i and predicted mean responses $\hat{Y}_i$.

Proposition 1.1. *Let the statistical model be $Y = X\beta + \varepsilon$, where X is a full-rank matrix*[1]*. The least squares estimator $\hat{\beta}$ is given by:*

$$\hat{\beta} = ({}^tXX)^{-1}\, {}^tXY.$$

[1] A matrix is full-rank if no column is linearly dependent on the others, i.e., the rank of the matrix equals the number of columns.

Proof. To find $\hat{\beta}$ that minimizes $\|Y - X\beta\|^2 = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$, note that $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ can be expressed as:

$$\langle Y - \hat{Y}, Y - \hat{Y}\rangle = {}^t(Y - \hat{Y})(Y - \hat{Y}),$$

where $\hat{Y} = X\beta$. Substituting this, we get:

$$\begin{aligned} Q(\beta) &= {}^t(Y - X\beta)(Y - X\beta) \\ &= ({}^tY - {}^t\beta\,{}^tX)(Y - X\beta) \\ &= {}^tYY - {}^t\beta\,{}^tXY - {}^tYX\beta + {}^t\beta\,{}^tXX\beta. \end{aligned}$$

Since ${}^tYX\beta = {}^t({}^t\beta\,{}^tXY) = {}^t\beta\,{}^tXY$, we can rewrite:

$$Q(\beta) = {}^tYY - 2\,{}^t\beta\,{}^tXY + {}^t\beta\,{}^tXX\beta.$$

To minimize Q, compute the derivative with respect to β:

$$\frac{\partial Q}{\partial \beta} = -2\,{}^tXY + 2\,{}^tXX\beta.$$

Setting this to zero gives:

$$-2\,{}^tXY + 2\,{}^tXX\beta = 0 \implies {}^tXX\beta = {}^tXY \implies \hat{\beta} = ({}^tXX)^{-1}\,{}^tXY.$$

To verify that this solution is a minimum, compute the second derivative:

$$\frac{\partial^2 Q}{\partial \beta^2} = 2\,{}^tXX.$$

Since ${}^tXX$ is positive definite, $Q(\beta)$ has a unique minimum at $\hat{\beta}$. □

Proposition 1.2. *If the assumptions in (1.2) about the residuals (errors) are satisfied, and if $\hat{\beta}$ is the least-squares estimator of β, then:*

1. *$\hat{\beta}$ is an unbiased estimator of β.*

2. *$\hat{\beta}$ has a variance-covariance matrix given by $\mathbb{V}\left(\hat{\beta}\right) = \sigma^2\left({}^tXX\right)^{-1}$.*

Proof. 1. Let us calculate $\mathbb{E}\left(\hat{\beta}\right)$:

$$\mathbb{E}\left(\hat{\beta}\right) = \mathbb{E}\left(\left({}^tXX\right)^{-1}{}^tXY\right) = \left({}^tXX\right)^{-1}{}^tX\mathbb{E}\left(Y\right) = \left({}^tXX\right)^{-1}{}^tXX\beta = \beta.$$

2. Replacing $\hat{\beta}$ by $({}^tXX)^{-1t}XY$ and Y by $X\beta + \varepsilon$, we get:

$$\hat{\beta} - \beta = ({}^tXX)^{-1t}X(X\beta + \varepsilon) - \beta.$$

Expanding this expression gives:

$$\hat{\beta} - \beta = ({}^tXX)^{-1t}XX\beta + ({}^tXX)^{-1t}X\varepsilon - \beta.$$

By simplifying, we find:

$$\hat{\beta} - \beta = \beta + ({}^tXX)^{-1}\ {}^tX\varepsilon - \beta = ({}^tXX)^{-1t}X\varepsilon.$$

Since the transpose of $\hat{\beta} - \beta$ is given by:

$$(\hat{\beta} - \beta)^t = {}^t\varepsilon X({}^tXX)^{-1},$$

we can write the variance-covariance matrix of $\hat{\beta}$ as:

$$\mathbb{V}(\hat{\beta}) = \mathbb{E}\left[(\hat{\beta} - \beta)\left(\hat{\beta} - \beta\right)^t\right].$$

By substituting $\hat{\beta} - \beta = ({}^tXX)^{-1}\ {}^tX\varepsilon$, we obtain:

$$\mathbb{V}(\hat{\beta}) = \mathbb{E}\left[({}^tXX)^{-1}\ {}^tX\varepsilon^t\varepsilon X({}^tXX)^{-1}\right].$$

Rearranging terms, we have:

$$\mathbb{V}(\hat{\beta}) = ({}^tXX)^{-1}\ {}^tX\mathbb{E}(\varepsilon^t\varepsilon)X({}^tXX)^{-1}.$$

Under the assumption that ε follows a normal distribution with zero mean and covariance matrix $\sigma^2 I_n$ (where I_n is the identity matrix of size n), we know that:

$$\mathbb{E}(\varepsilon^t\varepsilon) = \sigma^2 I_n.$$

Thus, substituting:

$$\mathbb{V}(\hat{\beta}) = ({}^tXX)^{-1t}X(\sigma^2 I_n)X({}^tXX)^{-1}.$$

Since ${}^tXI_nX = {}^tXX$, we get:

$$\mathbb{V}(\hat{\beta}) = \sigma^2({}^tXX)^{-1t}XX({}^tXX)^{-1}.$$

By simplifying further:

$$\mathbb{V}(\hat{\beta}) = \sigma^2({}^tXX)^{-1}.$$

□

1.4.5 Prediction of the Mean Response

Once $\hat{\beta}$ is determined, the experimenter is often interested in using the model to predict the mean response at any given point (where no experiment has been performed). This is crucial when modeling aims, for example, to identify experimental conditions that could maximize (or minimize) the response of interest. The prediction of the mean response at the point $x \in \mathbb{R}^k$ is given by:

$$\hat{Y}(x) =^t f(x)\beta$$

where $f(x) \in \mathbb{R}^p$ is a regression vector, constructed identically to the rows of the matrix X. Knowing the value of the predicted mean response at the point x, the quality of this prediction is quantified using the following result:

Proposition 1.3. *The quality of the prediction $\hat{Y}(x) =^t f(x)\beta$ at the point $x \in \mathbb{R}^k$ is measured by:*

$$\mathbb{V}(\hat{Y}(x)) = \sigma^2 f(x)^T({}^tXX)^{-1}f(x).$$

It can be observed that this error on the calculated response (or predicted response) depends on four factors:

- *The experimental error on the measured responses,*
- *The position of the point x in the study domain,*
- *The set of points that were used to establish the model coefficients, i.e., the experimental design itself,*
- *The postulated model chosen to interpret the results (through the coefficient calculation matrix and the residual variance).*

Proof. We have:

$$\mathbb{V}(\hat{Y}(x)) = \mathbb{V}({}^tf(x)\hat{\beta}) = {}^tf(x)\mathbb{V}(\hat{\beta})f(x) = \sigma^2\,{}^tf(x)({}^tXX)^{-1}f(x)$$

since $\mathbb{V}(\hat{\beta}) = \sigma^2({}^tXX)^{-1}$. □

1.4.6 Prediction Variance Function

The error in the measured responses depends on the nature of the experimentation, the precision of the technology used, the care, skill of the experimenter, and many other factors for which the experimenter is responsible. These factors do not depend on the theory of experimental designs but on practical experimentation [18]. To separate this experimental component from the one that depends on the theory, we introduce the prediction variance function $d^2(\hat{Y})$:

$$d^2(\hat{Y}) =^t f(x)({}^tXX)^{-1}f(x).$$

By taking the square root of the variance function, we obtain the prediction error function:

$$d(\hat{Y}) = \sqrt{{}^tf(x)({}^tXX)^{-1}f(x)}.$$

1.5 Statistical Tests

1.5.1 The Multiple Correlation Coefficient R^2

The multiple correlation coefficient R^2 is a measure of the quality of fit of a multiple linear regression model to the data. It is calculated as follows:

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST} = \frac{\sum_{i=1}^n(\hat{Y}_i - \bar{Y})^2}{\sum_{i=1}^n(Y_i - \bar{Y})^2}$$

where:

- SSE (Sum of Squared Errors) is the sum of squared residuals of the model, defined as:
$$SSE = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2.$$
- SST (Total Sum of Squares) is the sum of squared differences between the observed values of the dependent variable, and it's mean:
$$SST = \sum_{i=1}^n (Y_i - \bar{Y})^2.$$

- SSR (Sum of Squares due to Regression) is the sum of squared deviations due to the regression:
$$SSR = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2.$$

1.5.2 Fisher's F Statistic

The Fisher formula is defined as [21]:

$$F = \frac{\frac{SSR}{p-1}}{\frac{SSE}{n-p}}$$

where:

- $(p-1)$ is the degree of freedom for SSR.
- $(n-p)$ is the degree of freedom for SSE.

If the Fisher F value is high, the calculated responses are significantly larger than the residual variance. To have significant coefficients, the Fisher F value must be high, indicating a low probability.

1.6 Some Common Experiment Designs

In this section, we introduce the main types of experiment designs, which can be categorized into two groups:

- Factorial designs.
- Response surface designs.

These two categories correspond to different potential objectives for using experiment design methods.

1.6.1 Factorial Designs

These designs aim to identify the most influential factors on a given response. The objective is not to establish precise relationships between factor variations and response variations. Factorial designs are discrete and orthogonal, where factors are discretized to take only a finite number of levels. Orthogonality is defined as follows:

Definition 1.3. *An experimental design is orthogonal if:*

- *For each level of a factor, all levels of any other factor appear the same number of times in the design.*
- *Each level of every factor appears the same number of times.*

In practice, factors in these designs typically take 2 to 5 levels. There are two types of factorial designs: complete factorial designs and fractional factorial designs.

1.6.1.1 Complete Factorial Designs

Complete factorial designs are orthogonal designs where factors take 2 or 3 levels, and all possible combinations of factor levels are evaluated. The design shown in Table 1.2 is a complete factorial design with three factors at 2, 2, and 3 levels, respectively. For k factors with 2 levels, the complete design is denoted as 2^k. For 3 levels, it is denoted as 3^k. If there are k_1 factors with 2 levels and k_2 factors with 3 levels, the design is denoted as $2^{k_1}3^{k_2}$.

Tableau 1.2: Experiment Matrix $2^2 \times 3^1$

N°	A	B	C
1	1	1	1
2	1	1	2
3	1	1	3
4	1	2	1
5	1	2	2
6	1	2	3
7	2	1	1
8	2	1	2
9	2	1	3
10	2	2	1
11	2	2	2
12	2	2	3

1.6.1.2 Fractional Factorial Designs

In practice, complete factorial designs are rarely used due to the large number of experiments required (Table 1.3). Instead, fractional factorial designs are often employed, representing an orthogonal fraction of the complete design. This fraction retains only certain experiments from the complete design, while preserving the property of orthogonality. Among fractional factorial designs, we find [22]:

Tableau 1.3: Number of Trials in a Complete 2^k Design

Number of Factors	Number of Trials
2	4
4	16
5	32
8	256

a. **Complete Fractional Designs 2^{k-m}:**

The approach offered by fractional factorial designs is to use the effect matrices of complete designs for k factors, such as $2^{k-1}, 2^{k-2}, \cdots$. The advantage of fractional factorial designs is evident, as the experimental workload is reduced by a factor of 2^m, given that $2^{k-m} = \frac{2^k}{2^m}$. For example, if we want to study three factors while performing only four trials, we ensure that the four trials are chosen such that the matrix X remains orthogonal. The four selected points are arranged as shown in Figure 1.8.

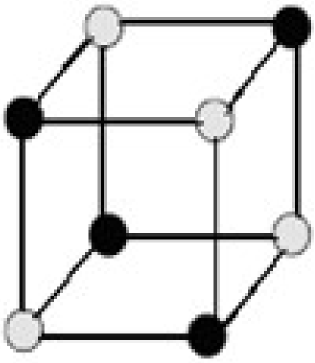

Figure 1.8: A complete factorial design 2^3 can be divided into two fractional factorial designs 2^{3-1}, one in black and one in gray.

b. **Plackett and Burman Designs:**

Plackett and Burman designs are restrictive designs that do not account for interactions. In these designs, factor effects are confounded with interactions. Interested readers can refer to [23] for more information on this type of design.

c. **Taguchi Designs:**

Taguchi [24] developed standard tables designed to solve most industrial problems. These standard tables take into account factor interactions, meaning the influence of one or more factors on others. Some second-order interactions, such as interactions between two factors, can be considered, while others cannot. The main orthogonal tables proposed by Taguchi include: $L_4(2^3)$, $L_8(2^7)$, $L_{16}(2^{15})$, $L_{12}(2^{11})$, $L_9(3^4)$, $L_{27}(3^{13})$, and $L_{32}(3^{31})$. Using these standard tables, along with their user-friendly and graphical presentation, it is easy to design custom experiment plans and interpret them.

Among these seven tables, the simplest is the $L_4(2^3)$ table, where "4" indicates that the table contains four rows, "2" corresponds to the two levels selected for each variable, and "3" represents the three factors. For instance, the following L_4 table is shown:

Tableau 1.4: Orthogonal Table $L_4(2^3)$

Factors Trials	1	2	3
1	1	1	1
2	1	2	2
3	2	1	2
4	2	2	1

1.6.2 Response Surface Designs

The objective of this category is not merely to prioritize the effects of different factors but to describe as accurately as possible the behavior of the response as a function of factor variations. The aim of this type of study is thus to achieve a modeling of the studied phenomenon based on experimentation. These designs help determine the values at which the input factors of a system should be adjusted to obtain one or more desired responses. They rely on the use of polynomial models. Numerous references on response surface designs exist [19, 25–28]. There are several types of such designs: full factorial designs with three levels, Box-Behnken designs, composite designs, optimal designs, and space-filling designs (such as Latin hypercube designs, Maximin designs, etc.).

1.6.2.1 Composite Designs

Composite designs allow for the calculation of a second-degree polynomial model, also known as a quadratic model. These designs consist of an initial part, which is either a fractional or full factorial design, one or more trials at the center of the study domain, and additional trials required to calculate the quadratic model. For two factors, the figure below illustrates the arrangement of experimental points for such a design:

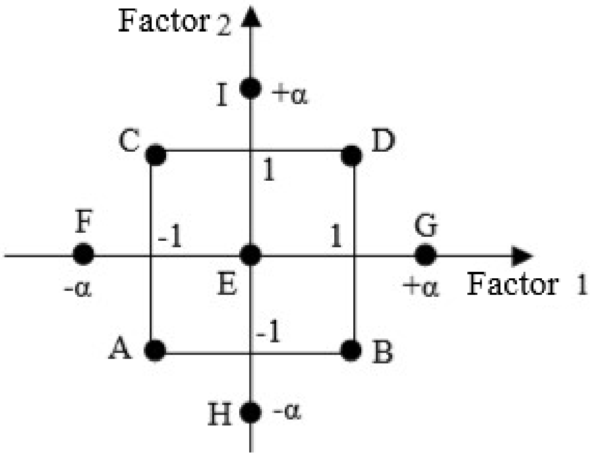

Figure 1.9: Composite design for the study of two factors

1.6.2.2 Box-Behnken Designs

Box and Behnken proposed these designs in 1960 [3] to establish second-degree models. All factors have three levels: -1, 0, and +1. These designs are easy to implement and possess the property of sequentially with respect to factors. This means that the study of the first *k* factors can be initiated while retaining the possibility of adding new factors without losing the results of already conducted trials. This property is valuable when a quick exploration of two or three most relevant factors is desired, while preserving the flexibility to study additional factors later.

1.6.2.3 D-Optimal Designs

D-optimal designs belong to a broader family known as alphabetic optimal designs (named after the optimality criteria satisfied by the designs: A, D, E, etc.). These designs are particularly well-suited for problems involving constraints, such as variation domain constraints (e.g., infeasible trials, reuse of previous trials, or previously conducted trials not aligned with the experimental points recommended by design theory), or constraints on the maximum number of trials. In such cases, the quality of the design deteriorates, especially with the loss of orthogonality. This reduces the precision of the estimators obtained. The challenge then becomes finding additional experiments within the constrained domain to allow for optimized estimation of the model, which is the objective of the D-optimal design [29].

1.7 Space-Filling Designs

These designs, often used in numerical experiments to study complex and non-linear models, employ various techniques to evenly fill the design space. Consequently, they do not rely on the concept of levels, eliminate the need for discretized parameters, and allow the experimenter to independently choose the sample size regardless of the number of parameters in the problem. Space-filling techniques are particularly suitable for constructing response surfaces, as for a given sample size N, the likelihood of empty regions far from any sample point—potentially causing inaccurate interpolation—is low. However, since these techniques do not use levels, evaluating main effects and interaction effects of parameters is less straightforward than in factorial experimental designs.

The properties often sought in these designs [30] include **fillness**, which maximizes the distance between the two closest points in the design; **independence**, which maximizes the determinant of the parameter matrix; and **uniformity** (discrepancy), which minimizes the distance between points. Various types of space-filling designs are adapted to meet these requirements.

Among space-filling designs, the following can be cited:

1.7.1 Latin Hypercube Designs

A Latin Hypercube Design (LHD) [31] with n trials is an experimental design where each factor has the same number of levels n, and each level is used exactly once per factor. The levels are distributed in a balanced manner, such that each column of the experimental design is a random sample without replacement from $1, 2, \ldots, n$. For illustration, a Latin Hypercube sampling in two dimensions divides the interval of each variable into $n = 5$ subintervals of equal size. For each variable and subinterval, a point is generated according to a uniform distribution, as shown in Figure 1.10.

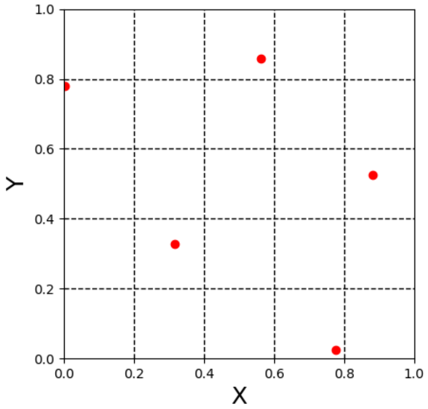

Figure 1.10: Five points resulting from Latin Hypercube sampling in two dimensions.

1.7.2 Low-Discrepancy Sequences

In the previous section, we discussed designs with good projection distributions but not necessarily optimal space-filling. Here, we present designs aimed at optimally filling the space and examine their projection properties. These sequences of points, used in the Monte Carlo method under the name quasi-Monte Carlo methods, are deterministically generated for uniform distribution in the experimental domain. Examples include Halton sequences [32], Sobol sequences [33], and Faure sequences [34]. An important concept underpinning the construction of most of these sequences is the definition of the radical inverse function, detailed below.

Definition 1.4. *Let $b \geq 2$ be an integer. Any integer $i \in \mathbb{N}$ can be uniquely decomposed in base b:*

$$i = \sum_{s=0}^{m} a_s b^s$$

where $a_s \in \{0, 1, \ldots, b-1\}$.

Using this decomposition, the radical inverse function in base b is defined as:

$$\phi_b(i) = \frac{a_0}{b} + \frac{a_1}{b^2} + \cdots + \frac{a_m}{b^{m+1}}$$

$$\text{where } m = \begin{cases} 1 + \lfloor \log_b(i) \rfloor, & \text{if } i \neq 0 \\ \\ 1, & \text{otherwise} \end{cases}$$

The sequence $C_b = \{x^0, x^1, \ldots, x^{n-1}\}$, where $x^i = \phi_b(i)$, is called the Van der Corput sequence in base b [35].

1.7.2.1 Halton Sequences

Halton sequences are the $d \geq 1$-dimensional extension of Van Der Corput sequences, which are their one-dimensional counterpart. The idea behind generating Halton sequences is to use a different base for each dimension.

Definition 1.5. *A Halton sequence $H_{b_1,b_2,\dots,b_d} = \{x^0, x^1, \dots, x^{n-1}\}$ in bases $b_1, b_2, \dots, b_d$ is defined as:*

$$x^i = (\phi_{b_1}(i), \phi_{b_2}(i), \dots, \phi_{b_d}(i)) \in [0,1]^d$$

where $b_1, b_2, \dots, b_d$ are positive integers that are pairwise coprime.

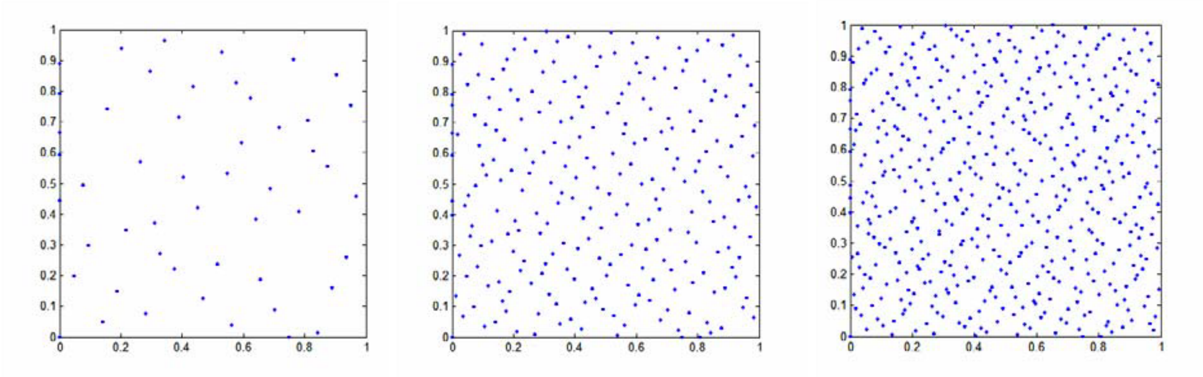

Figure 1.11: The first 50, 250, and 500 points of a Halton sequence with bases 2 and 3.

1.7.2.2 Sobol Sequences

These sequences are designed to distribute points within the space while minimizing the distance between each observation. Their construction is relatively complex and relies on linear recurrences generated from primitive polynomials over the finite field $\mathbb{Z}_2 = \{0, 1\}$. These sequences are described as quasi-random because the coordinates of each successive point can be determined from the previous one, and so on.

Definition 1.6. *A polynomial $p(t)$ of degree s of the form $t^s + u_1 t^{s-1} + \cdots + u_s$ is said to be primitive over the field $\mathbb{Z}_2$ if it is irreducible over $\mathbb{Z}_2$ and the smallest integer i such that it divides $t^i - 1$ (or $t^i + 1$) equals $2^s - 1$.*

The notion of irreducibility means verifying that the polynomial $p(t)$ is not divisible by any other polynomial of lower degree. The smallest positive integer i such that $p(t)$ divides $t^i - 1$ is also known as the order of the polynomial.

A primitive polynomial of degree s must have the terms 1 and t^s, as well as an odd number of terms.

Definition 1.7. *A Sobol′ sequence $S = \{x^0, x^1, \ldots, x^{n-1}\}$ in one dimension is defined as follows:*

$$x^i = \frac{1}{2^m}\left(\bigoplus_{k=1}^{m} a_k l_k\right)$$

where $(a_1, a_2, \ldots, a_m)$ is the binary representation of i and

$$m = \begin{cases} 1 + \lfloor \log_b(i) \rfloor, \text{ if } i \neq j \\ 1, \text{ otherwise} \end{cases}$$

The symbol $\oplus$ denotes addition in $\mathbb{Z}_2$ (modulo 2).

The l_k for $k > s$ are obtained using the following recurrence relation:

$$l_k = 2u_1 l_{k-1} \oplus 2^2 u_2 l_{k-2} \oplus \cdots \oplus 2^{s-1} u_{s-1} l_{k-s+1} \oplus 2^s u_s l_{k-s}$$

where the u_k are the coefficients of a primitive polynomial $t^s + u_1 t^{s-1} + \cdots + u_s$ over $\mathbb{Z}_2$, and the numbers $l_1, \ldots, l_s$ must be odd integers such that $1 \leq l_k \leq 2^k$ for $k = 1, \ldots, s$.

To construct a Sobol′ sequence in dimension d, one simply selects d distinct primitive polynomials.

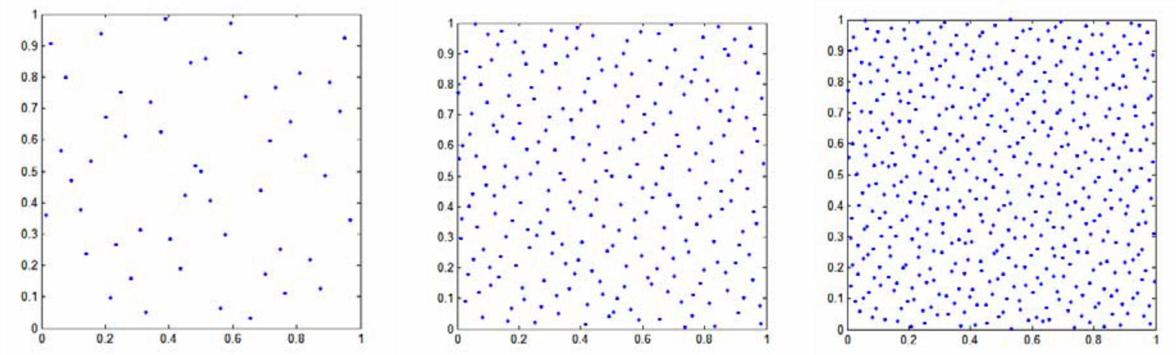

Figure 1.12: The first 50, 250, and 500 points of a Sobol sequence in two dimensions.

1.7.2.3 Faure Sequences

The Faure sequences are defined based on the inverse radical function ϕ_b, and a Pascal generator matrix $\mathcal{C} = (c_{lk})$, given by:

$$c_{lk} = C_{l-1}^{k-1} = \begin{cases} \dfrac{(l-1)!}{(k-1)!(l-k)!} \text{ if } k \leq l \\ 0 \text{ otherwise,} \end{cases} \qquad \forall l, k \in \mathbb{N}^*.$$

Definition 1.8. *Let $b \geq d$ be a prime number. The Faure sequence $F = \{x^0, x^1, \ldots, x^{n-1}\}$ in dimension d is defined as follows:*

$$x_j^i = \phi_b\left(\mathcal{C}_i^{j-1}\right)$$

where $\mathcal{C}^{j-1} \equiv C_{l-1}^{k-1}(j-1)^{l-k} \mod b$ is the generator matrix for the j^{th} dimension of a Faure sequence in dimension d.

Note. It is recommended to choose the smallest prime number b greater than or equal to d to achieve a uniform distribution.

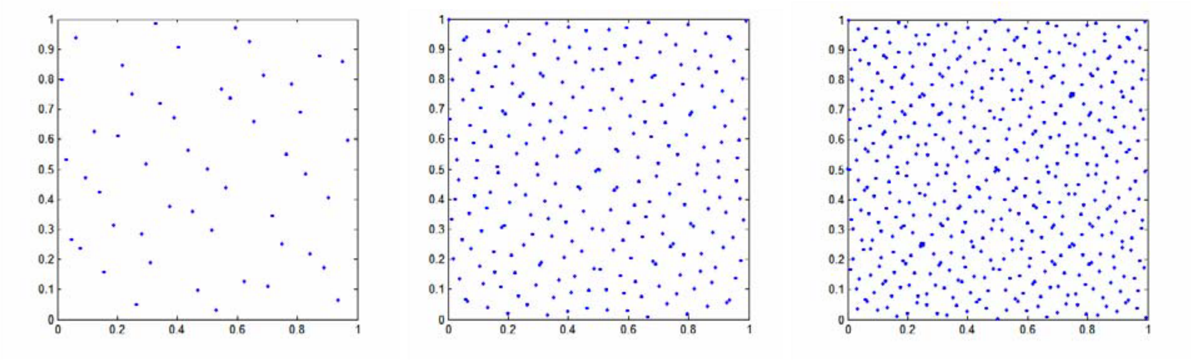

Figure 1.13: The first 50, 250, and 500 points of a Faure sequence in two dimensions.

Note that Faure sequences are considered superior to both Halton and Sobol sequences.

1.8 Optimality Criteria for Experimental Design Plans

Evaluating the quality of the structure of a set of points from a database or experimental design plan requires using quantitative criteria. There are numerous criteria to assess the quality of an experimental design. Generally, for response surface plans, we aim to optimally place the experimental points to minimize the error in the predicted responses. Among these criteria are orthogonality, quasi-orthogonality, D-criterion, A-criterion, and others [12]. In contrast, for space-filling designs, we focus on the uniformity of point distribution using criteria such as discrepancy, distance, and coverage. Our study specifically focuses on space-filling designs, which is why we define the following criteria:

- **The Distance Criterion** *Mindist* [36]: This criterion aims to maximize the minimum distance between two points in the design. The larger the value of this criterion, the more distant the points are from each other.

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

where $d(x_i, x_j)$ is the Euclidean distance between the points x_i and x_j.

- **The Coverage Criterion** *Cov* [37]: This criterion measures the gap between the points of the plan and those of a regular grid. It is zero for a regular grid. The objective is to minimize coverage to approach a regular grid, ensuring the space is filled.

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

  where $\sigma_i = \min_{i \neq j} d(x_i, x_j)$ and $\bar{\sigma} = \frac{1}{n} \sum_{i=1}^{n} \sigma_i$.

  For a regular grid, $\sigma_0 = \sigma_1 = \cdots = \sigma_n$, so $cov = 0$.

  In the same context, we can use the ratio R, defined by:

  $$R = \frac{\max \sigma_i}{\min \sigma_i}$$

  For a regular grid, $R = 1$. Therefore, the closer R is to 1, the closer the points are to those of a regular grid.

- **The Discrepancy Criterion** *Disc* [38]: Discrepancy measures the gap between the empirical distribution function of the points and the uniform distribution law. The lower the discrepancy, the more uniformly the points are distributed. There are various discrepancy measures, and we consider the L_2-norm discrepancy:

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

Conclusion

In summary, the method of experimental design is a set of complementary techniques that help users determine the experiments to perform, as well as understand and exploit the results obtained. The tools used in this method are primarily based on statistical and algebraic foundations. The developments presented in this chapter have outlined the principles, foundations, and analytical possibilities of experimental design methods. This analytical method is perfectly suited for studying systems thanks to its multiple aspects.

Chapter 2

Point Processes

Point processes are essential for analyzing spatial data where objects are scattered across a region. Examples include forest data, microscopic analyses, scene images, physical particles, or geological sketches, among others. These objects are often represented as point maps. This representation necessitates analyzing the distribution and spatial patterns influenced by heterogeneity and interactions between points. For instance, in sustainable agriculture, studying pesticide spot patterns in soil reveals clusters and open spaces. Point processes thus enable a deeper understanding and interpretation of these complex patterns across various fields.

This chapter aims to present some fundamental aspects of the theory of point processes, accompanied by illustrative examples. For more detailed discussions, it is recommended to consult [5,39,40], which serve as the primary references for the elements presented here.

2.1 Basic Definitions and Notations

2.1.1 Mathematical Context

We are interested in unordered sets of points within a given space $\mathcal{X} \subset \mathbb{R}^p$.

Definition 2.1. *A **configuration** is a countable, unordered set of points in $\mathcal{X}$:*

$$x = \{x_1, x_2, \dots, x_n, \dots\}.$$

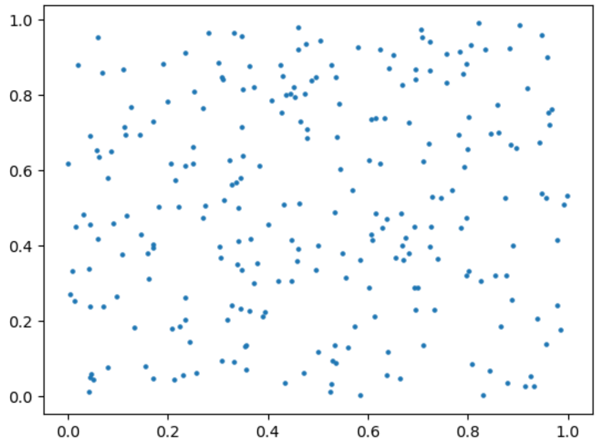

Figure 2.1: A set of points, or configuration, in $\mathcal{X} = [0,1]^2$.

To study random configurations, constraints must be imposed on the space under consideration. We work with a space $\mathcal{X}$ equipped with a metric d, such that $(\mathcal{X}, d)$ is complete and separable. This metric defines a topology and a Borel σ-algebra on $\mathcal{X}$. In practice, $\mathcal{X}$ is often a compact subset of $\mathbb{R}^p$ with the Euclidean distance.

Definition 2.2. *A configuration $x \in \mathcal{X}$ is said to be* ***locally finite*** *if, for every bounded Borel set $A \subseteq \mathcal{X}$, it contains a finite number of points, denoted $N_x(A)$.*

The set of all locally finite configurations is denoted by N^{lf}.

This leads to the definition of a point process:

Definition 2.3. *A* ***point process*** *on $\mathcal{X}$ is a mapping X from a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ into N^{lf}, such that for every Borel set $A \subseteq \mathcal{X}$, $N(A) = N_X(A)$ is a (finite) random variable.*

This provides the following refinement:

Definition 2.4. *If the space $\mathcal{X}$ is bounded or if $N_X(\mathcal{X})$ is almost surely finite, then the point process is called a* ***finite point process****.*

The realizations of a point process X are random configurations of points such that, for every Borel set $A \subseteq \mathcal{X}$, the number of points in A is a random variable. This implies that a point process is a random variable taking values in the measurable space $(N^{lf}, \mathcal{N}^{lf})$, where $\mathcal{N}^{lf}$ is the smallest σ-algebra for which the mapping $x \mapsto N_x(A)$ is measurable for every bounded Borel set $A \subseteq \mathcal{X}$. The probability measure induced on $\mathcal{N}^{lf}$ is called the law of X.

2.1.2 Law of a Point Process

The law of a point process X should ideally be the push forward measure of $\mathcal{P}$ on $\mathcal{N}^{lf}$ via the mapping X. However, since $\mathcal{N}^{lf}$ is defined based on the measurability of mappings $x \mapsto N_x(A)$ for Borel sets $A \subseteq \mathcal{X}$, the probabilistic analog of the law in the context of point processes is the collection of joint distributions of the vectors $(N(A_1), \ldots, N(A_m))$, where A_i are bounded Borel sets.

Definition 2.5. *The family of finite-dimensional distributions (fidis) of a point process X on a complete and separable metric space $(\mathcal{X}, d)$ is the collection of joint distributions of the vectors $(N(A_1), \ldots, N(A_m))$, for every finite vector $(A_1, \ldots, A_m)$ of bounded Borel sets $A_i \subseteq \mathcal{X}$, where $i = 1, \ldots, m$ for any $m \in \mathbb{N}$.*

The importance of this definition is justified by the following theorem of Daley and Jones [4]:

Theorem 2.1. *The law of a point process X on a complete and separable metric space $(\mathcal{X}, d)$ is entirely determined by its finite-dimensional distributions (fidis).*

Therefore, two-point processes with the same finite-dimensional distributions have the same law.

2.1.2.1 Marked Point Processes

This type of point process was described by Daley [41].

Point processes are the simplest examples of stochastic geometry, but it is often necessary to simulate processes involving more complex objects. To describe objects instead of points, one can simply assign a "mark" to each point that characterizes the object’s properties (e.g., size, orientation, shape, etc.). This results in what is called a "marked point process" or "object process" [4].

Definition 2.6. *Let $(\mathcal{X}, d)$ and $(\mathcal{K}, d')$ be two complete and separable metric spaces. A **marked point process**, with positions in $\mathcal{X}$ and marks in $\mathcal{K}$, is a point process on $\mathcal{X} \times \mathcal{K}$ such that the unmarked process of points is itself a well-defined point process.*

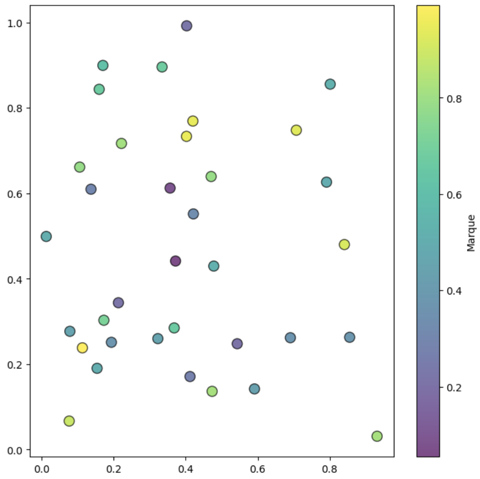

Figure 2.2: Realization of a marked point process on $\mathcal{X} = [0,1]^2$ and $\mathcal{K} = [0,1]$.

2.1.3 Simple Point Processes

We now consider the potential multiplicity of points in a configuration. In practice, it is rare to work with point processes where multiple points are located at the exact same position. Let N_s^{lf} denote the set of locally finite configurations x containing only distinct points, i.e., $N_x(\{y\}) \in \{0,1\}$ for all $y \in x$.

To verify that N_s^{lf} is $\mathcal{N}^{\mathrm{lf}}$-measurable, note that since $\mathcal{E}$ is separable, it can be covered by a countable union of open balls $B(x_i, 2^{-j})$ with arbitrarily small radii. Consequently, N_s^{lf} can be expressed as:

$$N_s^{\mathrm{lf}} = \bigcup_{j=1}^{\infty} \left\{ \omega \in \Omega : N(B(x_i, 2^{-j})) \in \{0,1\} \right\} \in \mathcal{N}^{\mathrm{lf}}.$$

Definition 2.7. *A point process X is said to be **simple** if it takes values in N_s^{lf} almost surely.*

Simple point processes are advantageous because their law can be fully determined by their void probabilities:

$$v(A) = \mathcal{P}(N(A) = 0),$$

for a sufficiently large class of sets $A \subset \mathcal{X}$.

Theorem 2.2. *The law of a simple point process X on a complete and separable metric space $(\mathcal{X}, d)$ is entirely determined by the void probabilities of bounded Borel sets $A \subset \mathcal{X}$.*

The proof can be found in [5].

2.2 Finite Point Processes

Most point processes used in practice are observed within a bounded region. This region may be determined by the application itself or chosen deliberately to limit the size of the domain of interest. In any case, the realizations of the point process almost surely contain a finite number of points.

An important theoretical reason for focusing on this class of point processes is that the notion of density is challenging to establish for processes that are not finite.

2.2.1 Construction

To construct a finite point process, the following elements can be used:

1. A discrete probability distribution $(p_n)_{n\in\mathbb{N}}$ for the number of points.
2. A family of symmetric probability densities $j_n(x_1,\ldots,x_n)$, for $n \in \mathbb{N}$, on the configuration space $\mathcal{X}^n$ for the positions of the points.

The second point assumes that $\mathcal{X}$ is equipped with a Borel measure $\nu(\cdot)$ so that the densities j_n can be defined with respect to the product measure $\nu^n(\cdot)$.

We can construct a point process X as follows: Let $N(\mathcal{X})$ be a random variable with distribution $(p_n)_{n\in\mathbb{N}}$, and condition on the events $N(\mathcal{X}) = n$. For each case, let $(X_1,\ldots,X_n) \in \mathcal{X}^n$ be a random vector with distribution $j_n(\cdot,\ldots,\cdot)$. The symmetry condition is necessary since a configuration does not depend on the order in which its points are listed. Verifying whether we can legally transition between ordered vectors and unordered configurations is important. This point will be useful in later proofs where we move between configurations and n-tuples, and vice versa.

Let N^f denote the set of finite configurations. Define $\mathcal{N}^f$ as the smallest σ-algebra for which the mappings $x \mapsto N_x(A)$ (for bounded Borel sets A) are measurable. Define the following subsets of N^f:

$$N_n = \{x \in N^f : N_x(\mathcal{X}) = n\},$$

and associate with each subset N_n an σ-algebra $\mathcal{N}^f_n$, which is the trace of $\mathcal{N}^f$ on N_n.

Now consider the functions $f_n : \mathcal{X}^n \to N_n$ that map n-tuples to configurations of n points. The functions f_n are measurable with respect to the Borel σ-algebra, and due to their invariance under permutations, they are measurable with respect to the symmetric Borel σ-algebra $\mathcal{B}_s(\mathcal{X}^n)$ in $\mathcal{X}^n$. By assumption, for all n, $j_n(\cdot,\ldots,\cdot)$ is a permutation-invariant density and is thus $\mathcal{B}_s(\mathcal{X}^n)$-measurable.

Consider the function $i_n : N_n \to \mathbb{R}$ defined on configurations of n points by:

$$i_n(x) = \frac{1}{n!} \sum_{\varphi} j_n(\varphi(x)),$$

where the sum is over all permutations φ that map a configuration to one of its n-tuples. Since there are $n!$ distinct permutations, we have $i_n \circ f_n = j_n$.

To show that i_n is measurable:

Let $A \subset \mathcal{X}$ be a bounded Borel set. We must verify that:

$$i_n^{-1}(A) \in \mathcal{N}_n^f.$$

Since f_n is surjective:

$$i_n^{-1}(A) = f_n\left(f_n^{-1}(i_n^{-1}(A))\right) = f_n\left(j_n^{-1}(A)\right).$$

As j_n is measurable with respect to the symmetric σ-algebra, it suffices to show that $f_n(B) \in \mathcal{N}_n^f$ for all $B \in \mathcal{B}_s(\mathcal{X}^n)$.

Define the σ-algebra:

$$\mathcal{A}_n = \{B \in \mathcal{B}_s(\mathcal{X}^n) : f_n(B) \in \mathcal{N}_n^f\}.$$

The σ-algebra $\mathcal{A}_n$ contains all rectangles A_n, where A is a bounded Borel set. Since such rectangles generate $\mathcal{B}_s(\mathcal{X}^n)$, it follows that $\mathcal{A}_n = \mathcal{B}_s(\mathcal{X}^n)$.

Thus, $f_n(B) \in \mathcal{N}_n^f$ for all $B \in \mathcal{B}_s(\mathcal{X}^n)$, establishing a correspondence between functions on the configuration space N_n and symmetric functions on $\mathcal{X}^n$.

2.2.2 Reference Point Processes

2.2.2.1 Binomial Point Process

Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^p$ with strictly positive volume $\mu(\mathcal{X})$.

A **binomial point process** is defined as the union:

$$X = \{X_1, \ldots, X_n\},$$

of a fixed number n of independent and uniformly distributed points $X_1, \ldots, X_n$ in $\mathcal{X}$. Since, $\mathcal{P}(X_i = X_j) = 0$ for all $i \neq j$, X is simple. Furthermore, as $\mathcal{P}(N(\mathcal{X}) = n) = 1$, the binomial process is finite with:

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

The points X_i are uniformly distributed, so:

$$j_n(x_1, \ldots, x_n) = \frac{1}{\mu(\mathcal{X})^n}.$$

This shows that j_n are invariant under permutation.

The binomial process is named for the fact that, for any Borel set $A \subset \mathcal{X}$:

$$N(A) = \sum_{i=1}^{n} \mathbb{1}_A(X_i)$$

follows a binomial distribution with parameters n and $\frac{\mu(A)}{\mu(\mathcal{X})}$.

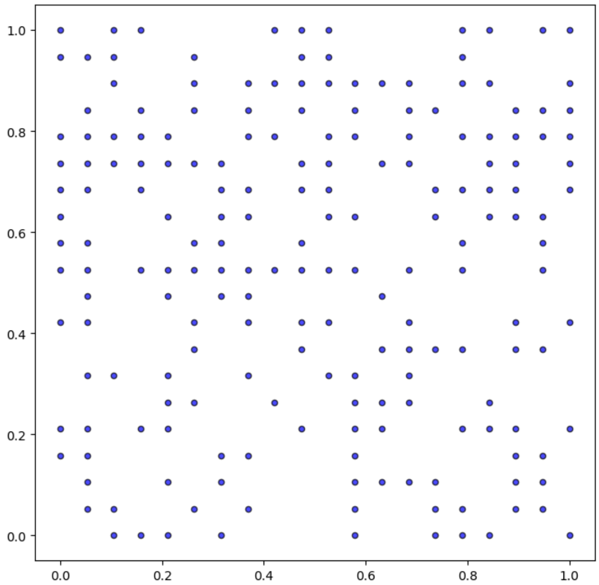

Figure 2.3: Realization of a binomial point process.

2.2.2.2 Poisson Point Process

Poisson point processes are among the most well-known and easy-to-use models for representing spatial randomness, such as raindrops on asphalt. In practice, we first test whether a point configuration is random or follows a certain pattern (dependence, clustering, etc.). We will present the results leading to a rigorous definition of the Poisson point process, starting with binomial processes, and define both homogeneous and inhomogeneous Poisson processes. These processes are elementary as they illustrate the concept of uniformity [42].

Definition 2.8. *Let $\nu(\cdot)$ be a Borel measure on a complete and separable metric space $(\mathcal{X}, d)$ such that $\nu(\mathcal{X}) > 0$ and $\nu(A) < \infty$ for every bounded Borel set $A \subset \mathcal{X}$. (Such a measure is called locally finite.)*

*A point process X on $\mathcal{X}$ is called a **Poisson point process** with intensity measure $\nu(\cdot)$ if:*

***P1.** $N(A)$ follows a Poisson distribution with mean $\nu(A)$ for every bounded Borel set $A \subset \mathcal{X}$.*

***P2.** For k disjoint Borel sets $A_1, \ldots, A_k$, the random variables $N(A_1), \ldots, N(A_k)$ are independent.*

If $\nu(\cdot)$ is nonatomic [1], the Poisson process is simple. It is finite if $\nu(\mathcal{X}) < \infty$.

When $\mathcal{X} = \mathbb{R}^d$, a Poisson process is called **homogeneous** if the intensity measure is $\lambda_\mu(\cdot)$, where λ is the Lebesgue measure and $\mu > 0$ is a positive parameter. This parameter is referred to as the intensity of the process.

Property **P2** can be interpreted as total spatial independence, indicating that the occurrence of points in any given region is completely independent of what happens outside that region. The following theorem illustrates this idea by stating that the points of a Poisson process behave randomly without interacting with each other [4,5].

Let X be a Poisson point process on a complete and separable metric space $(\mathcal{X}, d)$ with intensity measure $\nu(\cdot)$. Let $A \subset \mathcal{X}$ be a bounded Borel set. Then, conditional on $\{N(A) = n\}$, the process X restricted to A follow the law of a binomial process of n independent points uniformly distributed according to ν on A.

The proof relies on the following result:

For a Borel set $B \subset A$, the void probability of B, given that there are n points in A, is:

$$\nu_A(B) = \mathcal{P}(N(B) = 0 \mid N(A) = n) = \frac{\mathcal{P}(N(B) = 0, N(A \setminus B) = n)}{\mathcal{P}(N(A) = n)}.$$

Using property **P2**, we know that $N(A)$ and $N(A \setminus B)$ are independent. Additionally, $N(A)$, $N(B)$, and $N(A \setminus B)$ follow a Poisson distribution by property **P1**. Thus:

$$\nu_A(B) = \left(\frac{\nu(A \setminus B)}{\nu(A)}\right)^n,$$

[1] A measure is nonatomic or diffuse if, for any measurable set A with $\nu(A) > 0$, there exists a measurable subset $B \subset A$ such that $0 < \nu(B) < \nu(A)$. In other words, no single point has a positive measure.

which matches the void probabilities of B for n independent points uniformly distributed with respect to ν. Since B is arbitrary, the application of Theorem 2 completes the proof.

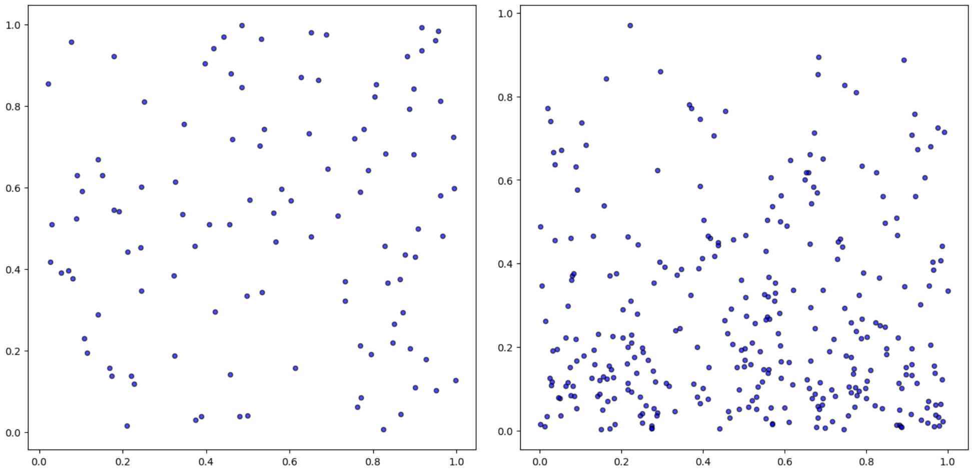

Figure 2.4: Left: A realization of a homogeneous Poisson point process with intensity $\lambda = 100$. Right: A realization of an inhomogeneous Poisson point process with $\lambda(x, y) = 600e^{-3y}$ on $[0, 1]^2$.

2.2.3 Finite Point Processes Defined by a Density

This section aims to construct point process models using their probability density (Radon-Nikodym derivative) with respect to a reference Poisson process.

Non-Finite Point Processes

The following lemma demonstrates the difficulty of defining densities for non-finite point processes.

Lemma 2.1. *Let X_λ and X_μ be homogeneous Poisson processes on $\mathbb{R}^p$, defined on the same probability space $(\Omega, \mathcal{A}, \mathcal{P})$, with intensity measures λ and μ, respectively.*

If $\lambda \neq \mu$, then the law of X_λ is not absolutely continuous with respect to the law of X_μ.

Proof. Consider the family $(B_n)_n$ of closed balls centered at the origin, with radii chosen such that the volume of B_n equals n. For any $\nu > 0$, define:

$$E_\nu = \left\{ \omega \in \Omega : \frac{N_{X_\nu}(\omega)(B_n)}{n} \to \nu \right\}.$$

For $i \in \{1, 2, \dots\}$, let $L_i = B_i \setminus B_{i-1}$. The sets L_i are disjoint, and by the definition of Poisson processes, under the law of X_λ, the random variables $N(L_i)$ are independent and follow a Poisson distribution with mean λ. By the strong law of large numbers:

$$\frac{1}{n}N(B_n) = \frac{1}{n}\sum_{i=1}^{n} N(L_i) \to \lambda \quad \text{a.s.}$$

Thus:

$$\mathcal{P}(X_\lambda \in E_\lambda) = 1, \quad \mathcal{P}(X_\mu \in E_\mu) = 1.$$

This is a form of the spatial strong law of large numbers: as the balls grow, realizations of the point process X_ν increasingly conform to the constraint that the "average number of points per unit volume" is ν.

Since E_λ and E_μ are disjoint (because $\lambda \neq \mu$), it follows that the law of X_λ is not absolutely continuous with respect to the law of X_μ. □

Recall that the definition of a finite point process requires that the total number of points in the space is almost surely finite. If we replace the above homogeneous processes with finite point processes, the average number of points per unit volume does not converge to the intensity as the region of interest grows, but rather to zero. Therefore, the above proof does not apply in this case.

Finite Point Processes

In this section, let $(\mathcal{X}, d)$ be a complete and separable metric space, and let $\pi(\cdot)$ denote the distribution of a Poisson process on $\mathcal{X}$ with a finite and nonatomic intensity measure $\nu(\cdot)$.

Let $p : N^f \to [0, \infty)$ be a positive measurable function defined on the space of finite point configurations such that:

$$\int_{N^f} p(x) d\pi(x) = 1, \tag{2.1}$$

then $p(\cdot)$ is a probability density and defines a point process X on $\mathcal{X}$. Since the dominant Poisson process is finite and simple, the process X is also finite and simple, represented as the union of the families N_n of configurations with n points:

$$N^f = \bigcup_{n=0}^{\infty} N_n.$$

The volume of N_n is $\nu(\mathcal{X})^n/n!$. The factor $n!$ accounts for the fact that $\mathcal{X}^n$ is ordered while N^f is not. Thus:

$$\lambda(N^f) = \sum_{n=0}^{\infty} \frac{\nu(\mathcal{X})^n}{n!} = e^{\nu(\mathcal{X})}.$$

From these considerations and the definition of the Poisson process, we deduce that:

The law of the total number of points of a process, defined by its density $p(\cdot)$ is given by the family $(p_n)_{n\in\mathbb{N}}$:

$$p_n = \frac{e^{-\nu(\mathcal{X})}}{n!} \int_{\mathcal{X}} \cdots \int_{\mathcal{X}} p(\{x_1, \dots, x_n\}) d\nu(x_1) \cdots d\nu(x_n).$$

Conditional on $\{N(\mathcal{X}) = n\}$, the n random points have a joint density with respect to $\nu^n(\cdot)$:

$$j_n(x_1, \dots, x_n) = \frac{p(\{x_1, \dots, x_n\})}{\int_{\mathcal{X}} \cdots \int_{\mathcal{X}} p(\{x_1, \dots, x_n\}) d\nu(x_1) \cdots d\nu(x_n)}.$$

2.3 Markov Point Processes

The intuitive definition of Markov point processes is relatively simple: they are finite point processes defined by a density where local knowledge depends only on a certain neighborhood.

Markov point processes are widely used in various applications. In image processing, their main advantage lies in their ease of computational implementation. Historically, they have also been extensively used in statistical physics, particularly under the name of Gibbs point processes. The distinguishing feature of Gibbs processes is that their densities are expressed in an energy form using interaction potentials between the points of a realization of the process. For more details, see [39,43–45].

We briefly present some definitions and properties of Markov point processes here.

Consider a symmetric and reflexive relation $\sim$ on $\mathcal{X}$. Two points u and v in $\mathcal{X}$ are said to be neighbors if $u \sim v$. For example, on $\mathcal{X} = \mathbb{R}^p$, we can define the neighborhood relation:

$$u \sim v \text{ if } d(u,v) \leq R,$$

where $d(u,v)$ is the distance between u and v, and R is a fixed distance threshold.

Definition 2.9. *The neighborhood $\partial(A)$ of a set $A \subseteq \mathcal{X}$ is defined as:*

$$\partial(A) = \{x \in \mathcal{X} : \exists a \in A \text{ such that } x \sim a\}.$$

Ripley and Kelly [46] give the following definition of a Markov point process:

Definition 2.10. *Let $(\mathcal{X}, d)$ be a complete and separable metric space, $\nu(\cdot)$ a finite nonatomic Borel measure, and $\pi_\nu(\cdot)$ the law of a Poisson point process with intensity measure $\nu(\cdot)$.*

Let X be a point process on $\mathcal{X}$ defined by its density $p(\cdot)$ with respect to $\pi_\nu(\cdot)$.

*Then X is a **Markov point process** under the symmetric and reflexive relation $\sim$ on $\mathcal{X}$ if, for all $x \in N_f$ with $p(x) > 0$, the following conditions hold:*

1. *$p(y) > 0$ for all $y \subseteq x$.*
2. *For all $u \in \mathcal{X}$, the ratio $\frac{p(x \cup \{u\})}{p(x)}$ depends only on u and on:*

$$\partial(\{u\}) \cap x = \{x_i \in x : u \sim x_i\}.$$

Condition (1) means that if a configuration can occur, then all its sub-configurations can also occur.

The ratio $\frac{p(x \cup \{u\})}{p(x)}$, called the **Papangelou conditional intensity** [47], represents the probability density that a point u is present, given that the configuration x is realized elsewhere. This expresses a local Markov property: the behavior of a point u with respect to the entire configuration depends only on its neighbors.

The following theorem expresses the density of a Markov process in a more practical form. First, we define:

Definition 2.11. *Let $\sim$ be a symmetric and reflexive neighborhood relation on $\mathcal{X}$. A configuration $x \in N^f$ is called a **clique** if all elements of x are neighbors, i.e.:*

$$\forall u, v \in x,\ u \sim v.$$

By convention, the empty configuration is also a clique.

We can then state the following theorem by Ripley and Kelly, which is equivalent to the Hammersley-Clifford theorem [46] for point processes:

Theorem 2.3. *A point process density $p : N^f \to [0, \infty)$ is Markovian under a neighborhood relation $\sim$ if and only if there exists a measurable function $\phi : N^f \to [0, \infty)$ such that:*

$$p(x) = \prod_{\text{cliques } y \subset x} \phi(y),$$

for all $x \in N^f$.

An alternative formulation involves taking a product over all subsets of x, imposing $\phi = 1$ for subsets that are not cliques.

2.3.1 Examples

Consider the space $\mathcal{X} = \mathbb{R}^p$ and a bounded subset $K \subset \mathcal{X}$ equipped with the Lebesgue measure $\lambda(\cdot)$. Densities can be defined with respect to a Poisson point process on K with intensity measure $\lambda(\cdot)$. This process is both simple and finite, which is necessary for defining point process densities.

Example 2.3.1 (Intensity Parameters). Consider processes with a density $h(\cdot)$ of the form:

$$h(x) = \alpha \cdot \beta^{n(x)},$$

where $\beta > 0$, and $n(x)$ represents the number of points in the configuration x. Such a density is integrable, and α is the normalization constant. The parameter β adjusts the intensity of the process, which is then $\beta\lambda(\cdot)$.

It is noteworthy that β acts as a scaling parameter: one can change the bounded subset of interest K by defining a new set K' such that $\lambda(K') = \beta \cdot \lambda(K)$.

Example 2.3.2 (Strauss Process). Consider the density of the form:

$$p(x) = \alpha \cdot \beta^{n(x)} \prod_{i<j} g(x_i, x_j), \quad \text{with } x = \{x_1, \ldots, x_{n(x)}\},$$

where the function g is defined as:

$$g(x_i, x_j) = \begin{cases} \gamma, & \text{if } d(x_i, x_j) < r, \\ 1, & \text{if } d(x_i, x_j) \geq r, \end{cases}$$

with $0 \leq \gamma \leq 1$ and $r > 0$. Such a process is called the **Strauss process**. It is a Markov process under the neighborhood relation $u \sim v \iff d(u, v) < r$.

Using the Hammersley-Clifford theorem extended to point processes, this density can be simplified as:

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

where $s(x)$ represents the number of pairs of points in relation to the configuration x.

we examine the impact of Parameter γ:

- For $\gamma = 1$, the process reduces to a Poisson point process on the bounded Borel set K with intensity $\beta\lambda(\cdot)$.
- For $\gamma \in (0, 1)$, the process exhibits repulsion between nearby points based on the proximity relation.

- For $\gamma = 0$, the process becomes a Hard Core Process, where the density forbids the presence of neighboring points in the configuration.

The case $\gamma > 1$, which was of interest to Strauss [48], allows for attraction between points in a realization. However, in this case, the density is not integrable unless multiplied by a factor such as $1_{\{n(x) \leq n_0\}}$ for some fixed integer n_0.

There are two perspectives to understand the non-integrality of this density:

1. **Combinatorial Perspective** (Van Lieshout and Baddeley [49]): Intuitively, as more points are added in a small space, the density increases rapidly.

2. **Exponential Families** (Geyer [50]): This approach provides a formalism to study the limiting behaviors of the process using exponential families.

2.4 Marked Markov Point Processes

Marked point processes are point processes where marks are attached to each event. These are particularly useful in applications where measurements are taken at each observation, or where points are of different types. For example, In forestry, researchers mapping a pattern of trees may record the height or trunk diameter of each tree, or species labels if multiple species are present, In materials science or image analysis, a pattern of objects may be modeled as a point process of kernels (e.g., center of gravity or another characteristic point) marked by shape descriptors.

The definition of a Markov point process extends naturally to marked point processes [6, 46]. Let $\mathcal{X}$ and $\mathcal{K}$ be complete and separable metric spaces, and let $m(\cdot)$ be a probability distribution on the Borel σ-algebra of $\mathcal{K}$. A suitable dominant Poisson process in this context has locations following a Poisson process on $\mathcal{X}$, independently marked by labels distributed according to m.

Definition 2.12. *Let $(\mathcal{X}, d)$ and $(\mathcal{K}, d')$ be complete and separable metric spaces, $\nu(\cdot)$ a finite nonatomic Borel measure on $\mathcal{X}$, $m(\cdot)$ a probability distribution on the Borel σ-algebra of $\mathcal{K}$, and $\pi_{\nu \times m}(\cdot)$ the distribution of a Poisson process on $\mathcal{X} \times \mathcal{K}$ with intensity measure $\nu \times m$.*

Let Y be a marked point process with positions in $\mathcal{X}$ and marks in $\mathcal{K}$, specified by a density $p(\cdot)$ with respect to $\pi_{\nu \times m}(\cdot)$. Then Y is a ***marked Markov point process*** *under the symmetric and reflexive relation $\sim$ on $\mathcal{X} \times \mathcal{K}$ if for all y such that $p(y) > 0$:*

a) $p(z) > 0$ for all $z \subseteq y$.

b) For all $(u,l) \in \mathcal{X} \times \mathcal{K}$, the ratio $\frac{p(y\cup\{(u,l)\})}{p(y)}$ depends only on (u,l) and:

$$\partial\{(u,l)\} \cap y = \{(x,k) \in y : (u,l) \sim (x,k)\}.$$

The Hammersley-Clifford theorem also applies to marked point processes [6, 46]. Thus, a probability density function $p(\cdot)$ defines a marked Markov point process with respect to the relation $\sim$ on $\mathcal{X} \times \mathcal{K}$ if and only if it can be factorized as:

$$p(y) = \prod_{\text{cliques } z \subseteq y} \phi(z),$$

for all $y \in N^f$, where the product is restricted to $\sim$-cliques $z \subseteq y$, and $\phi : N^f \to [0,\infty)$ is the interaction function.

2.4.1 Examples

Example 2.4.1. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^p$, and consider the pair interaction process with a finite number of marks on $\mathcal{X} \times \{1,\ldots,K\}$ defined by [6]

$$p(y) = \alpha \prod_{(u,k)\in y} \beta_k \prod_{(u,k),(v,l)\in y} \gamma_{kl} \left(\|u-v\|\right)$$

for some $\alpha > 0$, intensity parameters $\beta_k > 0$, and measurable interaction functions $\gamma_{lk} : [0,\infty[\to [0,\infty[$. The reference distribution is that of a Poisson process with unit rate, with locations marked by a uniformly distributed label.

Without loss of generality, assume that $\phi(r) = 1$ for $\gamma_{lk} > 0$ for some range parameters $\gamma_{lk} = 1$. The process is then Markovian with respect to the brand-dependent relation

$$(u,k) \sim (v,l) \quad \text{if} \quad \|u-v\| \leq r_{kl}$$

2.5 Random Tessellations

Tessellations, or mosaics, are partitions of the plane or space into polygons and polyhedra. Among the various tessellations, Voronoi tessellations hold a special place due to their significance and wide range of applications. These geometric structures divide a space into regions based on proximity to a set of reference points.

This section focuses exclusively on Voronoi tessellations, a class of random tessellations where each region, or Voronoi cell, is defined as the set of points in the plane or space that are closest to a particular point among a given set of points. Voronoi tessellations appear in many natural and applied fields, including crystalline structures, crack patterns, and foam arrangements.

We will explore some geometric properties of Voronoi tessellations, presenting precise mathematical definitions for both planar and spatial cases. Generalizations to higher dimensions will also be discussed [51, 52].

2.5.1 Voronoi Tessellations on $\mathbb{R}^p$

Let us consider the affine plane $\mathbb{R}^p$. Let x be a finite set of n points in the plane, and the elements $\{x_i\}_{i=1\cdots n}$ of x are called centers, sites, or seeds. The Voronoi region or Voronoi cell associated with a seed x_i from S is the set of points that are closer to x_i than to any other point in x:

$$\vartheta_x\left(x_i\right) = \{s \in \mathbb{R}^p \mid \forall q \in x, \|s - x_i\| \leq \|s - q\|\}$$

where $\|s - x_i\|$ denotes the distance between the point s and the seed x_i.

Let $H(x_i, q)$ denote the half-plane containing p delimited by the perpendicular bisector of the segment $[x_i q]$, we then have:

$$H(x_i, q) = \left\{s \in \mathbb{R}^2 \mid \|s - x_i\| \leq \|s - q\|\right\}$$

Thus, we can write:

$$\vartheta_x\left(x_i\right) = \bigcap_{q \in x \setminus \{x_i\}} H(x_i, q)$$

In two dimensions, it is easy to plot these partitions. We base this on the fact that the boundary between the Voronoi cells of two distinct seeds lies on the perpendicular bisector that separates these two seeds. Indeed, points on this bisector are equidistant from the two seeds, so it cannot be claimed that they lie in one or the other Voronoi cell. For a set of seeds, the Voronoi diagram is thus constructed by determining the perpendicular bisectors of each pair of seeds. A point on a bisector belongs to a Voronoi boundary if it is equidistant from at least two seeds and there is no shorter distance to another seed from the set.

Figure 2.5 shows examples of Voronoi tessellations for 10 and 20 points on $[0, 1]^2$.

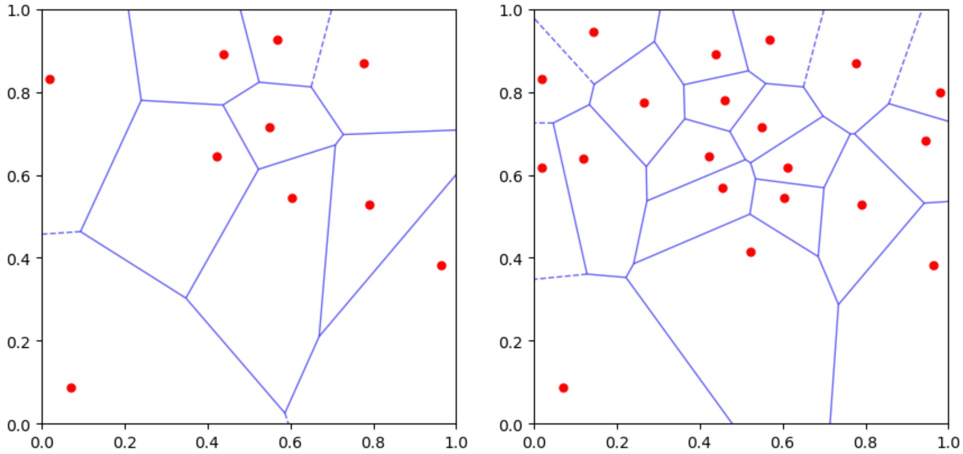

Figure 2.5: Example of Voronoi tessellations generated for 10 and 20 points on $[0,1]^2$.

We can generalize the notion to an Euclidean space E equipped with the Euclidean distance d. Let x be a finite set of n points in E. The definition becomes:

$$\vartheta_x(x_i) = \{s \in E \mid \forall q \in x, d(s,x_i) \leq d(s,q)\}$$

For two points a and b in x, the set $\Pi(a,b)$ of points equidistant from a and b is an affine hyperplane (a subspace of affine dimension 1). This hyperplane is the boundary between the set of points closer to a than to b and the set of points closer to b than to a:

$$\Pi(a,b) = \{s \in E \mid d(s,a) = d(s,b)\}$$

We denote by $H(a,b)$ the half-space bounded by this hyperplane containing a. It contains all points closer to a than to b. The Voronoi region associated with a is then the intersection of $H(a,b)$ where b ranges over $S \setminus \{a\}$:

$$H(a,b) = \{s \in E \mid d(x,a) \leq d(x,b)\}$$

$$\vartheta_x(a) = \bigcap_{b \in S \setminus \{a\}} H(a,b)$$

2.5.2 Generalization of the Voronoi Diagram

To solve certain problems, Shamos [53] introduced the concept of the Voronoi diagram of a set of points A (a subset of x), denoted $V(A)$, defined by:

$$\mathrm{V}(A) = \{s \in E \mid \forall x_i \in A, \forall q \in x \setminus A, d(s,x_i) < d(s,q)\}$$

Thus, $V(A)$ is the set of points that are closer to each point in A than to any point not in A.

Let $H(i,j)$ denote the half-plane delimited by the perpendicular bisector of the segment $[ij]$ and containing i, then we have:

$$\mathrm{V}(A) = \bigcap_{i \in A, j \in x \setminus A} H(i,j)$$

The generalized Voronoi regions are therefore convex, but they may be empty. Shamos later defined Voronoi diagrams of order k ($1 \leq k < \mathrm{card}(x)$) as the union of the generalized Voronoi cells formed by all subsets of k points:

$$\mathrm{V}_k(x) = \bigcup_{A \subset x, \mathrm{card}(A) = k} \mathrm{V}(A)$$

The regions $\mathrm{V}(A)$ form a partition of $\mathrm{V}_k(x)$.

Shamos also defined the "Voronoi diagram of the farthest points" (farthest-point Voronoi diagram). This diagram is constructed by reversing the direction of the inequality:

$$\overline{\vartheta_x}\,(x_i) = \{s \in E \mid \forall q \in x, d(s, x_i) \geq d(s,q)\}$$

The point x_i is obviously not in the cell $\bar{\vartheta}_x\,(x_i)$, but is opposite relative to the "center" of the set: x_i is the point in x furthest from $\bar{\vartheta}_x\,(x_i)$.

The diagram of the farthest points is entirely determined by the convex hull of x. It does not contain a closed cell.

Thus, the set of points furthest from a point x_i is the set of points that are closer to other points in x:

$$\bar{\vartheta}_x\,(x_i) = \bigcup_{q \in \complement_x(\{p\})} \vartheta_x\,(q)$$

Thus, the diagram of the farthest points is identical to $\mathrm{V}_{n-1}(x)$,
where $n = \mathrm{card}(x)$.

2.5.3 Properties

Voronoi regions are convex polytopes as intersections of half-spaces. The set of such polygons partitions E, and is the Voronoi partition corresponding to the set x.

Theorem 2.4. *Let v be a point in the plane. It is a vertex of a Voronoi polygon if and only if* *it is the center of a circle passing through three seeds, and containing no other seed within its surface.*

Proof. The point v lies at the intersection of the cells $\vartheta_x(q)$, $\vartheta_x(r)$, and $\vartheta_x(s)$. It is thus equidistant from the points q, r, and s. Therefore, v is the center of the circumcircle of the triangle formed by q, r, and s.

If another seed were inside the circumdisk of the triangle qrs, then v would be closer to this point than to q, r, and s. Consequently, v would not lie in the cells $\vartheta_x(q)$, $\vartheta_x(r)$, and $\vartheta_x(s)$ as it would not be at a shorter distance from the points q, r, and s compared to this new seed. □

2.6 Conclusion

Point processes are the simplest examples of stochastic geometry, but it is often necessary to model processes involving more complex objects, hence the interest in introducing the concept of marked point processes. These can model many common situations with ease. Markovian point processes, in particular, allow for the consideration of the environment and possible dependencies between the various points in the experiment. We have provided the necessary definitions and propositions to model and more precisely describe these point distributions.

However, simulating these processes remains crucial for practical applications and in-depth analyses. We will explore in detail the methods for simulating these processes in Chapter 3.

Chapter 3

Monte Carlo Method by Markov Chain (MCMC)

Since their emergence in the mid-20th century, Monte Carlo methods have revolutionized the numerical solution of complex problems. Initially introduced to address differential equations in physics by Metropolis and Ulam in 1949 [54], these techniques quickly found broader applications as a general method for numerical integration. The proposal in 1953 by Metropolis et al. [55] of a sequential version of this method, further developed by Hastings in 1970 [9] to form the Markov Chain Monte Carlo (MCMC) methods, marked a pivotal turning point. The theoretical advances of the 1990s, combined with the increase in computing power, solidified MCMC as an essential tools in science.

The simulation of a probability distribution or a random variable has become an essential numerical solution in the absence of analytical solutions, whether combinatorial or continuous. This need led to the development of several sophisticated simulation algorithms, including the Metropolis-Hastings algorithm, which has several variants [56] distinguished by their efficiency and reliability. This chapter focuses on the fundamental principles of MCMC methods, particularly the Metropolis-Hastings algorithm, to explore their specific application in the simulation of point processes, providing an in-depth understanding of these techniques.

3.1 Monte Carlo Approach

In statistics, several situations require the computation of expectations of the form:

$$\pi(f) = \int_{\mathcal{X}} f(x)\pi(dx),$$

where π is a probability measure on the state space $\mathcal{X}$ and $f : \mathcal{X} \to \mathbb{R}$ is a function measurable with respect to π. In some cases, the analytical calculation of this expectation may be difficult or even impossible. In such situations, it is necessary to explore alternative methods to estimate $\pi(f)$ or at least obtain an approximation.

Since expectations are simply integrals with a particular form, it is possible to use numerical integration methods to approximate $\pi(f)$. Typically, these methods involve dividing the space $\mathcal{X}$ into rectangles and approximating the function using simpler functions (e.g., constant or linear functions) over these rectangles. However, this approach faces challenges when the dimension of $\mathcal{X}$ is high: the number of rectangles required to achieve a given level of precision in the approximation grows exponentially with the dimension. Furthermore, when the integration domain $\mathcal{X}$ is unbounded, this method may also encounter difficulties.

Monte Carlo methods, which are specifically designed to handle expectations, are generally less affected by these problems. They are particularly suitable for situations where traditional numerical integration methods encounter limitations.

3.1.1 The principle of the Monte Carlo method

The fundamental principle behind Monte Carlo methods relies on the law of large numbers. A law of large numbers is any result of the form:

$$\frac{1}{N}\sum_{n=1}^{N} f(x_n) \xrightarrow{c} \pi(f), \quad N \to \infty,$$

which means that the average of the function f taken over a sample $\{x_n\}_{n=1}^{N}$ converges, under certain conditions, to the expectation of this function taken under the distribution π for a mode of convergence $c \in \{\mathcal{P}, \text{p.s.}\}$. This sample average can, in turn, be viewed as the following expectation:

$$\frac{1}{N}\sum_{n=1}^{N} f(x_n) = \int_{\mathcal{X}} f(x)\hat{\pi}_N(dx) = \hat{\pi}_N(f), \tag{3.1}$$

where $\hat{\pi}_N$ is the empirical mass function of the Monte Carlo sample $\{x_n\}_{n=1}^N$ given by

$$\hat{\pi}_N(x) = \frac{1}{N}\sum_{n=1}^{N} \delta_{x_n}(x),$$

and where $\delta_y(\cdot)$ denotes the Dirac delta function at y. This means that the empirical measure $\hat{\pi}_N$ is used in place of π in the expectation. It becomes evident that the properties of the sample $\{x_n\}_{n=1}^N$ required for the Monte Carlo estimator $\hat{\pi}_N(f)$ to satisfy a law of large numbers are such that the empirical measure $\hat{\pi}_N$ provides a good approximation of the distribution π. The different Monte Carlo methods typically differ only in how the sample is produced.

A second property often sought in an estimator is a central limit theorem. This type of result shows that the asymptotic distribution of the estimator is a Gaussian distribution:

$$\frac{1}{\sqrt{2N}}\hat{\pi}_N(f) \xrightarrow{\mathcal{D}} \mathcal{N}(\pi(f), \sigma_f^2), \quad n \to \infty,$$

where $\xrightarrow{\mathcal{D}}$ denotes convergence in distribution and where σ_f^2 is the asymptotic variance of the estimator. When a central limit theorem is satisfied, it is possible to attach a Monte Carlo standard error to the point estimate $\hat{\pi}_N(f)$ given by $\hat{\sigma}_f/\sqrt{N}$, where $\hat{\sigma}_f^2$ is an estimate of σ_f^2, such that the sample variance is

$$\hat{\sigma}_f^2 = \frac{1}{N}\sum_{n=1}^{N} (f(x_n) - \hat{\pi}_N(f))^2.$$

Thus, it is possible to provide an assessment of the quality of the Monte Carlo estimate when such a result holds. Note that most of the terms used in Monte Carlo methods are also used more generally in statistics (estimator, standard error, sample, etc.). To distinguish the two concepts, the term "Monte Carlo" is often added after these terms. This distinction is especially relevant when the expectation $\pi(f)$ itself depends on a real sample from an experiment; the Monte Carlo sample is a collection of points in X and not the set of units in the experiment. Similarly, the Monte Carlo standard error does not correspond to the standard error of the sample mean.

3.1.2 Monte Carlo Algorithms

The simplest versions of the law of large numbers require that each element of the sample be a realization of independent and identically distributed random variables according to the target distribution π, which is denoted by $X_n \overset{i.i.d}{\sim} \pi$. The empirical distribution will therefore be well-representative of the target distribution since the sample is generated directly from π. This type of sampling, called i.i.d. sampling and described in Algorithm 2.1, is the standard Monte Carlo method. However, two problems often prevent the use of i.i.d. sampling.

```
Algorithm 1: Standard i.i.d. Monte Carlo Algorithm
  Data: Target distribution $\pi$ and Monte Carlo sample size $N$.
  Procedure
      for $n = 1$ to $N$ do
          Sample $X_n \overset{\text{i.i.d.}}{\sim} \pi$;
  Result: The sample $x_{1:N}$ and the Monte Carlo estimate (3.1).
```

First, it may happen that sampling directly from π is impossible. Sampling must be done by computer, where only uniform random numbers are typically available. The probability integral transform method allows one to transform a uniform distribution into a more complex distribution, but this transformation is not always known or possible. The important sampling method allows for bypassing this problem. We consider a second distribution q such that its support contains the support of π and from which it is possible to obtain an i.i.d. sample. Then, by observing the following identity,

$$\pi(f) = \int_{\mathcal{X}} f(x)\pi(x)dx = \int_{\mathcal{X}} f(x)\frac{\pi(x)}{q(x)}q(x)dx = q\left(f\frac{\pi}{q}\right),$$

we see that a Monte Carlo estimator of $\pi(f)$ can be obtained by the sample average of the function $f\pi/q$ over a sample from the distribution $q(\cdot)$:

$$\hat{\pi}_N(f) = \hat{q}_N\left(f\frac{f}{q}\right) = \frac{1}{N}\sum_{n=1}^{N} f(x_n)\frac{\pi(x_n)}{q(x_n)} = \int\limits_{\mathcal{X}} f(x_n)\frac{\pi(x_n)}{q(x_n)}\hat{q}_N(dx),$$

where $\hat{q}_N(\cdot) = \frac{1}{N}\sum_{n=1}^{N}\delta_{x_n}(\cdot)$ is the empirical distribution of the sample generated from q. This expression can also be interpreted as a weighted empirical distribution of π:

$$\hat{\pi}_N(x) = \frac{1}{N}\sum_{n=1}^{N} w(x_n)\delta_{x_n}(x), \quad w(x) = \frac{\pi(x)}{q(x)} \tag{3.2}$$

The ratio $w = \pi/q$ is called the likelihood ratio or the importance weight. In addition to not having to sample directly from π, this method has the advantage of potentially reducing the asymptotic variance of the Monte Carlo estimator compared to i.i.d. sampling. Although this is theoretically true, it is not always the practice case, as variance reduction requires knowledge of $\pi(|f|)$, which is typically unknown because we are already trying to estimate $\pi(f)$. However, preferential sampling requires the explicit calculation of $\pi(x)$ in the importance weights, which necessitates the exact expression of π. A second common issue in both i.i.d. and preferential sampling is that the distribution π may not be fully known. Indeed, certain situations arise where an expectation is sought for a distribution that is known only up to a proportionality constant.

For example, the context of Bayesian statistics often produces estimators given by a posterior expectation of the form:

$$\hat{\theta}(x) = \mathbb{E}\{\theta | X = x\} = \int \theta \pi(\theta|x)\, d\theta,$$

where the aim is to estimate a parameter θ using x and where π is the posterior distribution of θ given x:

$$\pi(\theta|x) = \frac{p(x|\theta)\pi_0(\theta)}{\int p(x|\theta)\pi_0(\theta)\, d\theta}, \tag{3.3}$$

where $\pi_0(\cdot)$ is the prior distribution on θ and p is the distribution of X given θ. The integral in the denominator of (3.3) is often difficult to evaluate, so the expression for $\pi(\theta|x)$ is not fully known.

More generally, this leads to the following situation: $\pi(x) \propto \tilde{\pi}(x)$. In this case, preferential sampling can be adapted to produce a valid Monte Carlo estimator. The importance weights are then known only up to a proportionality constant:

$$\tilde{w}(x) = \frac{\tilde{\pi}(x)}{q(x)} \propto \frac{\pi(x)}{q(x)} = w(x).$$

In this case, the empirical measure $\frac{1}{N}\sum_{n=1}^{N} \tilde{w}(x_n)\delta_{x_n}(x)$ is no longer a probability measure, since it does not integrate to 1 over $\mathcal{X}$ with respect to π. Therefore, this measure needs to be normalized in order to obtain the correct importance weights:

$$\hat{\pi}_N(x) = \frac{1}{N}\sum_{n=1}^{N} w(x_n)\delta_{x_n}(x), \quad w(x_n) = \frac{\tilde{w}(x_n)}{\sum_{n=1}^{N} \tilde{w}(x_n)}.$$

This corresponds to the following reformulation of $\pi(f)$, noting that $\tilde{\pi}(x) = \tilde{w}(x)q(x)$:

$$\begin{aligned}\pi(f) &= \int_{\mathcal{X}} f(x)\pi(x)dx \\ &= \int_{\mathcal{X}} f(x)\frac{\pi(x)}{q(x)}q(x)dx \\ &= \int_{\mathcal{X}} f(x)\tilde{w}(x)\frac{\pi(x)}{\tilde{\pi}(x)}q(x)dx \\ &= \int_{\mathcal{X}} f(x)\frac{\tilde{w}(x)q(x)}{\int_{\mathcal{X}} \tilde{\pi}(x)dx}dx \\ &= \int_{\mathcal{X}} f(x)\frac{\tilde{w}(x)q(x)}{\int_{\mathcal{X}} \tilde{w}(x)q(x)dx}dx\end{aligned}$$

Thus, the calculation of π is no longer necessary when the importance weights are normalized; Algorithm 2.2 details the procedure. Note that having $\tilde{w} = w$ (i.e., $\tilde{\pi} = \pi$) reduces to regular preferential sampling, as we then have:

$$\int_X \tilde{w}(x)q(x)\,dx = \int_X w(x)q(x)\,dx = \int_X \pi(x)\,dx = 1.$$

```
Algorithm 2: Monte Carlo Algorithm with Weighted Sampling
Data: Target distribution $\pi \propto \tilde{\pi}$, Monte Carlo sample size $N$, and instrumental
      density $q$ with support including the support of $\pi$.
Procedure
    for $n = 1$ to $N$ do
        (a) Sampling: Generate $X_n \sim q$ independently of $X_{1:n-1}$;
        (b) Weights: Compute the importance weight $\tilde{w}(x_n) = \frac{\tilde{\pi}(x_n)}{q(x_n)}$;
    2. Normalize the importance weights:;
        $w(x_n) = \frac{\tilde{w}(x_n)}{\sum_{n=1}^{N} \tilde{w}(x_n)} \quad n = 1 \cdots N;$
Result: The sample $x_{1:N}$ and the Monte Carlo estimate (3.2).
```

Another type of sampling avoids the problem of the target density being known only up to a proportionality constant. We again consider $\pi \propto \tilde{\pi}$ and an instrumental density q. Rejection sampling assumes that there exists a constant $M < \infty$ such that $\tilde{\pi} \leq Mq$, meaning that a multiple of q completely envelops π. A candidate $X \sim q$ is included in the Monte Carlo sample with probability $\tilde{\pi}(x)/Mq(x)$. The procedure (Algorithm 3) is repeated until the sample contains the required number of points.

Each of the accepted points in the sample will be distributed according to π; indeed, for a measurable set $B \in \mathcal{B}(X)$, we have

$$\mathcal{P}(X \in B \mid \text{accepted}) = \frac{\mathcal{P}(X \in B, \text{ accepted})}{\mathcal{P}(\text{accepted})} = \frac{\int_B \frac{\tilde{\pi}(x)}{Mq(x)} q(x)\,dx}{\int_{\mathcal{X}} \frac{\tilde{\pi}(x)}{Mq(x)} q(x)\,dx}$$

$$= \frac{\int_B \tilde{\pi}(x)\,dx}{\int_{\mathcal{X}} \tilde{\pi}(x)\,dx} = \int_B \pi(x)\,dx = \pi(B).$$

Note that the efficiency of rejection sampling heavily depends on the choice of q. Indeed, the probability of accepting a candidate is given by

$$\mathbb{P}(\text{accepted}) = \int_X \frac{\tilde{\pi}(x)}{Mq(x)} q(x)\,dx = \frac{1}{M} \int_X \tilde{\pi}(x)\,dx = \frac{1}{\tilde{M}}.$$

```
Algorithm 3: Monte Carlo Algorithm with Rejection Sampling
Data: Target distribution $\pi \propto \tilde{\pi}$, Monte Carlo sample size $N$, and instrumental
      density $q$ such that $\tilde{\pi} \leq Mq$ for some $M < \infty$.
Procedure
    for $n = 1$ to $N$ do
        Sampling. Generate $X \sim q$ and $U \sim \text{Uniform}[0,1]$;
        Acceptance. If $U < \frac{\tilde{\pi}(X)}{Mq(X)}$, accept $x_n = X$ and set $n = n + 1$;
Result: The sample $x_{1:N}$ and the Monte Carlo estimate (3.1).
```

Thus, on average, f_M candidates will be needed to produce a new point in the sample. When f_M is large, several iterations of the algorithm will be wasted. It is possible to recycle rejections using the Casella et al. [57] method, which can improve efficiency by reducing the variance of the estimates.

The Metropolis-Hastings algorithm, described in 3.4.1, also uses the acceptance/rejection concept to produce a Monte Carlo sample. The restrictions on the instrumental density in this algorithm are less severe, which helps avoid the previously mentioned problem. However, the Metropolis-Hastings algorithm requires the use of Markov chains. The next section will introduce these concepts, which are essential for understanding algorithms like Metropolis-Hastings.

3.2 Markov Chain

To discuss simulation using the MCMC (Markov Chain Monte Carlo) method, it is essential to understand the underlying principles that make these methods effective for generating realizations. MCMC methods exploit the properties of Markov chains to produce simulations that, after a sufficient number of iterations, provide a reliable approximation of the target distribution $\pi(.)$.

Markov chains are stochastic processes where the probability of transitioning from one state to another depends only on the current state, not on how the state was reached. The MCMC method relies on this property to construct a sequence of values that converge to the desired distribution.

Before diving into the application of MCMC methods, it is crucial to understand these properties of Markov chains, as well as the theories that underlie their functioning. This section provides a review of the necessary theory to analyze these methods, drawing on important works [58,59].

3.2.1 Basic Concepts

In this section, we consider a probability space $(\Omega, \mathcal{A}, \mathcal{P})$ and a space $\mathcal{X}$ equipped with its Borel sigma-algebra $\mathcal{B}(\mathcal{X})$, with random variables taking values in $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$. Let's start with a few definitions:

Definition 3.1. *A transition kernel is a function K defined on $\mathcal{X} \times \mathcal{B}(\mathcal{X})$ such that:*

1. *For all $x \in \mathcal{X}$, $K(x, \cdot)$ is a probability measure on $\mathcal{B}(\mathcal{X})$,*
2. *For all $A \in \mathcal{B}(\mathcal{X})$, $K(\cdot, A)$ is a measurable function.*

This definition allows us to give another one:

Definition 3.2. *A Markov chain is a sequence of random variables $(X_n)_{n\in\mathbb{N}}$ taking values in $\mathcal{X}$ such that, for all integers $k \in \mathbb{N}$ and all $A \in \mathcal{B}(\mathcal{X})$,*

$$\mathcal{P}(X_{k+1} \in A \mid X_0, \ldots, X_k) = \mathcal{P}(X_{k+1} \in A \mid X_k)$$

or equivalently,

$$\mathcal{P}(X_{k+1} \in A \mid X_0, \ldots, X_k) = \int_A K\left(x_k, dx\right)$$

In general, Markov chains are restricted to homogeneous Markov chains, meaning chains where the law of $(X_{t_1}, \ldots, X_{t_k})$ given x_0 is the same as the law of $(X_{t_1-t_0}, \ldots, X_{t_k-t_0})$ given x_0 for all k and for any $(k+1)$-tuple $t_0 \leq t_1 \leq \cdots \leq t_k$.

Some reminders are provided below to understand the use of the kernel K, as well as to introduce common notations:

$$\mathcal{P}_x\left(X_1 \in A_1\right) = K(x, A_1)$$

$$\mathcal{P}_x\left(X_1 \in A_1, X_2 \in A_2\right) = \int\limits_{A_1} K\left(y, A_2\right) K\left(x, dy\right)$$

$$\mathcal{P}_x\left(\left(X_1, \ldots, X_n\right) \in A_1 \times \cdots \times A_n\right) = \int\limits_{A_1} \cdots \int\limits_{A_n} K\left(y_{n-1}, A_n\right) K\left(y_{n-2}, dy_{n-1}\right) \cdots K\left(x, dy\right)$$

The following notations are related to transitions in n steps:

$$K^1(x, A) = K(x, A)$$

$$K^n(x, A) = \int_{\mathcal{X}} K^{n-1}(y, A) K(x, dy)$$

To complete this section of notations, the following explanations are provided:

- $\mathcal{P}$ denotes the law of the chain $(X_n)_n$ with $X_0 = x$,
- $\mathcal{P}_\mu(\cdot)$ denotes the law of the chain with an initial distribution $X_0 \sim \mu$,
- $\mathbb{E}_\mu[\cdot]$ denotes the expectation associated with the law $\mathcal{P}_\mu$.

Basic Properties of Markov Chains

Here, we will discuss some common properties of Markov chains, such as the Chapman-Kolmogorov equations, the weak Markov property, and the strong Markov property. These properties are generally intuitive and well-established, and for more details, one can refer to [59]. In this text, Robert and Casella also introduce the concept of a resolving kernel, a theoretical tool important for the proofs of the following propositions and theorems.

Invariant Measures (Stationarity)

Invariant measures are fundamental objects associated with Markov chains. We place them here in the context of recurrence notions.

Definition 3.3. *An invariant measure π on $\mathcal{B}(\mathcal{X})$ for the transition kernel $K(\cdot,\cdot)$ is defined as:*

$$\pi(B) = \int_{\mathcal{X}} K(x,B)\pi(dx) \quad \forall B \in \mathcal{B}(\mathcal{X}).$$

When the invariant measure is also a probability measure, it is called a stationary measure.

Reversibility

A Markov chain is said to be reversible with respect to π if the transition kernel K satisfies the following condition:

$$\int_A K(x,B)\pi(dx) = \int_B K(x,A)\pi(dx) \quad \forall A, B \in \mathcal{A}.$$

This condition implies that π is an invariant measure for P, and means that under the stationary distribution π, the probability of transitioning from A to B is the same as the probability of transitioning from B to A. Most simulation algorithms are designed to generate reversible Markov chains.

Green [60] proposed an alternative formulation of reversibility for a transition kernel K. This condition is given by:

$$\forall x \in A, \forall y \in B : \int_A \int_B \pi(dx)K(x,dy) = \int_B \int_A \pi(dy)K(y,dx)$$

Stopping Rules and Associated Quantities

Definition 3.4. *A stopping rule is any function $\xi(x_1,\ldots,x_n,\ldots)$ that takes values in $\mathbb{N}$ such that the events $\{\xi = n\}$ are measurable with respect to the sigma-algebra generated by $(X_0,\ldots,X_n)$.*

An important example of a stopping rule is as follows: for a set $A \subset \mathcal{X}$, the stopping time at A is defined as the first time n when the chain visits this set:

$$\tau_A = \inf\{n \geq 1 \mid X_n \in A\}.$$

By convention, $\tau_A = \infty$ if the chain never visits A. It is clear that the event $\{\tau_A = n\}$ depends only on the random variables $(X_0,\ldots,X_n)$.

In the same spirit, another random variable, the number of visits to A, is defined as:

$$\eta_A = \sum_{n=1}^{\infty} 1_A(X_n).$$

These random variables are of paramount importance in the study of Markov chains. In particular, the following associated quantities are crucial:

- $\mathbb{E}_x[N_A]$, which represents the average number of visits to A for the Markov chain starting from state x,
- $\mathcal{P}_x(\tau_A < \infty)$, which represents the probability of reaching set A in finite time starting from state x.

Irreducibility

Throughout this section, φ represents a measure on $\mathcal{X}$.

Definition 3.5. *Consider a Markov chain (X_n) with transition kernel K. This chain is called φ-irreducible if, for any set $A \subset \mathcal{X}$ such that $\varphi(A) > 0$, the following two (equivalent) properties hold:*

- *For every $x \in \mathcal{X}$, there exists n such that $K^n(x, A) > 0$.*
- *For every $x \in \mathcal{X}$, $\mathcal{P}_x(\tau_A < \infty) > 0$,*

where τ_A is the hitting time of the set A.

Intuitively, a chain is irreducible if every set that is non-negligible in the sense of φ is reachable in finite time with a positive probability.

Proposition 3.1. *The Markov chain (X_n) is φ-irreducible if and only if, for every $x \in \mathcal{X}$ and every $A \subseteq \mathcal{X}$ such that $\varphi(A) > 0$, we have:*

$$\mathbb{E}_x[\eta_A] > 0$$

An interesting property of irreducible chains, which we will not develop here, concerns the existence of a measure for which the chain is irreducible and which dominates (in the sense of absolute continuity) any measure for which the chain is also irreducible. This measure allows for a form of the converse of irreducibility: a negligible set in the sense of this measure is never reached in finite time, except possibly starting from a set of points that is itself negligible.

Atoms and Small Sets

This section provides tools for working with "continuous" Markov chains in a manner similar to discrete chains.

Definition 3.6. *A Markov chain (X_n) has an atom $\alpha \in \mathcal{B}(\mathcal{X})$ if there exists a measure $\nu(\cdot) > 0$ such that:*

$$K(x, A) = \nu(A) \quad \forall x \in \alpha \quad \forall A \in \mathcal{B}(\mathcal{X}).$$

If (X_n) is φ-irreducible when $\varphi(\alpha) > 0$, the atom is said to be accessible.

This notion of atom is interesting because ideally, we would like to work with "piecewise constant" kernels. However, such constancy is rarely encountered in practice. Therefore, the following minimization condition is often preferred, which leads to the notion of small sets:

Definition 3.7. *A set C is small if there exists $m \in \mathbb{N}^*$ and a measure ν_m such that, for all $x \in C$ and all $A \in \mathcal{X}$, we have:*

$$K^m(x, A) \geq \nu_m(A) > 0 \tag{3.4}$$

Atoms are thus a special case of small sets. Note that there are two different but similar concepts: sets can be "small" in English or "petite" in French. Fortunately, these notions coincide when the chain is aperiodic. Here, we call small sets those that are "small" in English. The following theorem, proved in [58], holds:

Theorem 3.1. *Let (X_n) be an φ-irreducible Markov chain. For any set $A \subset \mathcal{X}$ such that $\varphi(A) > 0$, there exists a small set $C \subseteq A$.*

Furthermore, $\mathcal{X}$ can be partitioned into a countable number of small sets.

It appears that atoms are often more interesting than small sets, as the kernel is constant. Using renewal time theory, one can show that if:

- the chain satisfies condition (3.4) for a set C,
- and furthermore $\mathcal{P}_x(\tau_C < \infty) = 1$ for all $x \in \mathcal{X}$,

then it is possible to construct a companion chain $\tilde{X}_n$ that has an atom using the concept of renewal time:

Definition 3.8. *A renewal time is a stopping rule τ such that $(X_\tau, X_{\tau+1}, \ldots)$ is independent of $(X_{\tau-1}, X_{\tau-2}, \ldots)$.*

To construct this companion chain, consider a small set C that satisfies (3.4) and such that $\mathcal{P}_x(\tau_C < \infty) = 1$ for all x. Then, the chain is modified by artificially introducing a renewal time: when the chain enters C, X_{n+1} is simulated using a measure $\nu(\cdot)$ with probability ϵ, and using $\frac{K(X_n,\cdot)-\epsilon\nu(\cdot)}{1-\epsilon}$ with probability $1-\epsilon$. This is allowed because C is small (a pair $(\nu(\cdot), \epsilon)$ can be found), and it allows for the same marginal law given X_n (Bernoulli's law is used). The sequence of renewal times is then defined as $\tau_j = \inf\{n > \tau_{j-1} \mid X_n \in C \text{ and } X_{n+1} \sim \nu\}$. The companion chain is then $\tilde{X}_n = (X_n, \omega_n)$ with $\omega_n = 1$ when $X_n \in C$ and X_{n+1} is generated according to $\nu(\cdot)$. The set $\widetilde{\alpha} = C \times \{1\}$ is then an atom of the chain $\tilde{X}_n$, whose sub-chain (X_n) is always a Markov chain with the transition kernel $K(x_n, \cdot)$.

Cycles and Aperiodicity

These concepts are defined for φ-irreducible Markov chains (X_n). Here is an initial, rather intuitive definition:

Definition 3.9. *The sets $A_1, \ldots, A_m$ of $\mathcal{B}(\mathcal{X})$ form an m-cycle of the chain (X_n) if:*

$$\begin{cases} \mathcal{P}(x, A_2) = 1 & \text{for } x \in A_1 \\ \vdots & \vdots \\ \mathcal{P}(x, A_m) = 1 & \text{for } x \in A_{m-1} \\ \mathcal{P}(x, A_1) = 1 & \text{for } x \in A_m \end{cases} \quad \text{and} \quad \varphi\left(\bigcup_{i=1}^{m} A_i\right)^c = 0;$$

From this, we deduce the notion of aperiodicity:

Definition 3.10. *A φ-irreducible chain is aperiodic if it possesses a cycle of length $m = 1$.*

Here is a more mathematical definition of a cycle:

Definition 3.11. *A φ-irreducible chain (X_n) has a cycle of length h if there exists a small set C, an associated integer k, and a probability measure ν_m (see the definition of a small set) such that h is the* gcd *(greatest common divisor) of the values:*

$$\{m \mid \exists \delta_m \text{ such that C is small for } \nu_m \geq \delta_m \nu_k\}$$

It should be noted that, in these definitions, small sets are used as substitutes for states when the state space is finite.

Proposition 3.2. *Let K be a transition kernel that is φ-irreducible. The period of a state $\mathcal{B}(\mathcal{X})$ does not depend on x; we then say that it is the period of the Markov chain, denoted $h(K)$ or simply h.*

Lemma 3.1. *If a state x has period $h(x)$, then there exists $k(x) \geq 0$ such that the set of loop lengths $R(x) = \{n \geq 0 \mid K^n(x, x) > 0\}$ contains all multiples $kh(x)$ for $k \geq k(x)$.*

The proofs of proposition 3.2 and lemma 3.1 are available in [61].

It is worth noting that, in these definitions, small sets are used in place of states when the state space is infinite.

Transience and Recurrence

In the following definitions, we consider φ-irreducible Markov chains (X_n):

Definition 3.12. *A set $A \subseteq \mathcal{X}$ is called recurrent if, for all $x \in \mathcal{X}$, we have $\mathbb{E}_x[\eta_A] = \infty$, where η_A represents the first return time to A. The set $A \subseteq \mathcal{X}$ is called uniformly transient if there exists a constant M such that $\mathbb{E}_x[\eta_A] < M$ for all $x \in \mathcal{X}$. The set A is called transient if it can be decomposed into a countable union of uniformly transient sets B_i:*

$$A = \bigcup_{i=1}^{\infty} B_i$$

The following theorem is particularly interesting because it illustrates the importance of the existence of an atom for an irreducible chain. Indeed, it shows that for an irreducible chain, the property of having an atom can be generalized to the entire state space:

Theorem 3.2. *Let (X_n) be an φ-irreducible Markov chain with an accessible atom α:*

(i) If α is recurrent, then every set $A \subseteq \mathcal{X}$ such that $\varphi(A) > 0$ is recurrent.

(ii) If α is transient, then $\mathcal{X}$ is transient.

Consequently, we can extend the notions of transience and recurrence to Markov chains as follows:

Definition 3.13. *A Markov chain (X_n) is recurrent if:*

(i) There exists a measure φ such that (X_n) is φ-irreducible.

(ii) For every set $A \subseteq \mathcal{X}$ such that $\varphi(A) > 0$, we have $\mathbb{E}_x[\eta_A] = \infty$ for all $x \in A$.

The chain is transient if:

(i) It is φ-irreducible.

(ii) $\mathcal{X}$ is transient.

To conclude this section, we provide the classification theorem for irreducible chains:

Theorem 3.3. *A φ-irreducible Markov chain is either transient or recurrent.*

However, there remains a question: how can we distinguish a transient chain from a recurrent chain? The following proposition answers this question.

Proposition 3.3. *If a set C with $\varphi(C) > 0$ satisfies the following condition:*

$$\mathcal{P}_x(\tau_C < \infty) = 1 \quad \forall\, x \in C$$

then the φ-irreducible chain is recurrent.

Remark:
The transition kernel K is called primitive (or regular) if and only if there exists an integer $t \geq 1$ such that all entries of K^t are strictly positive.

If a Markov chain (X_n) is φ-irreducible and possesses a stationary measure, it is said to be positive.

Proposition 3.4. *If the chain (X_n) is positive, then it is recurrent.*

The following theorem provides a kind of converse to this proposition. It guarantees the existence and uniqueness of an invariant measure for recurrent chains. However, it does not guarantee that the obtained measure is integrable, and thus it does not ensure that the chain is positive.

Theorem 3.4. *If (X_n) is a recurrent chain, then there exists a finite measure on $\mathcal{B}(\mathcal{X})$ that is invariant, unique up to a multiplicative factor.*

A final, crucial theorem for our application framework is as follows. It is proven in [58].

Theorem 3.5. *If a Markov chain (X_n) is φ-irreducible for a nonzero measure $\varphi(\cdot)$ on $\mathcal{B}(\mathcal{X})$, and if it has a stationary distribution $\pi(\cdot)$, then the chain is also π-irreducible.*

3.3 Ergodicity and Convergence

We now address the concepts that interest us most, particularly ergodicity [62]:

Definition 3.14. *For a positive Markov chain with an invariant probability measure $\pi(\cdot)$, an atom α is said to be ergodic if:*

$$\lim_{n\to\infty} \|K^m(\alpha,\alpha) - \pi(\alpha)\| = 0.$$

This definition of ergodicity concerns the convergence of one measure to another. Indeed, this part aims to study the eventual convergence of the distribution of a Markov chain to its stationary distribution. To do this, it is first necessary to define the relevant norm on the space of measures. We consider the total variation norm:

$$\|\mu_1 - \mu_2\|_{VT} = \sup_{|A|<1} |\mu_1(A) - \mu_2(A)|$$

The important theorem is the following:

Theorem 3.6. *If the Markov chain (X_n) is aperiodic and positive, then:*

$$\lim_{n\to\infty} \left\| \int_{\mathcal{X}} K^m(x,.)\, v\,(dx) - \pi(\cdot) \right\|_{VT} = 0$$

for any initial distribution v.

This theorem also applies to initial distributions that are Dirac's. In particular, it guarantees that, regardless of the initial state of the chain, convergence to the stationary distribution is ensured. This is the theorem that gives practical importance to recurrence.

However, it is important to note that these convergence results are given in terms of $K^m(\cdot,\cdot)$, the transition kernel of the chain.

3.3.1 Control of Convergence

Controlling the convergence of $\|K^m(x,\cdot) - \pi\|$ to 0 is a central and complex issue. This control is crucial because it ensures that π is correctly simulated by K^m for sufficiently large m. Many theoretical results address this question. Some results use lower bounds for K on reduced sets of $\mathcal{X}$. In particular, the contraction coefficient in a finite state space is a useful tool for controlling this convergence [58].

3.3.2 Geometric and Uniform Ergodicity

Geometric ergodicity is characterized by:

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

where $M(x)$ is π-integrable and $\lambda < 1$. Ergodicity is said to be uniform if M can be chosen as a finite constant, independent of x.

Contraction Coefficient

In finite state space, the contraction coefficient for a transition kernel K is given by [63]:

$$C(P) = \frac{1}{2} \max_{x,y \in \mathcal{X}} \|K(x,\cdot) - K(y,\cdot)\|_{TV}$$

Lemma 3.2. *[64] Let ν and μ be two distributions, and P and Q be two transition kernels. Then:*

$$\|\mu P - \nu P\|_{TV} \leq \|\mu - \nu\|_{TV} \cdot C(P),$$

and

$$C(PQ) \leq C(P) \cdot C(Q).$$

In particular,

$$\|\mu P - \nu P\| \leq \|\mu - \nu\| \quad \text{and} \quad \|\mu P - \nu P\| \leq 2C(P).$$

Furthermore, if K is primitive, then $C(P) < 1$. From this result, if we choose $\mu = \pi$, the invariant distribution of K, we obtain:

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

Thus, as $m \to \infty$, the chain is uniformly ergodic if K is primitive.

3.4 Monte Carlo by Markov Chains

In section 3.2, we studied the properties of Markov chains and observed that certain conditions on the transition kernel K allow us to guarantee convergence results. Thus, a Markov chain can be used in a Monte Carlo simulation to estimate an expectation $\pi(\cdot)$. Throughout the following, we will denote the Markov kernel by P.

Unlike the Monte Carlo methods described in section 3.1, where the values generated in the simulation were independent, Markov Chain Monte Carlo (MCMC) methods produce values sequentially using a Markov chain, thus introducing a sequential dependence between the simulation outcomes. The general procedure is described in

Algorithm 4.

```
Algorithm 4: General MCMC Algorithm
Data: Target distribution $\pi$, Markov transition $P$, and the number of Monte
      Carlo simulations $N$.
Procedure
    1. Initialization: Initial value of the chain $x_0$;
    for $i = 0$ to $N - 1$ do
        Generate the new state of the chain
            $X_{i+1} \mid X_i = x_n \sim P(\cdot \mid x_i)$;
Result: $x_N$.
```

Markov Chain Monte Carlo (MCMC) algorithms include several methods for simulating complex distributions. Among these methods, the Metropolis-Hastings algorithm stands out for its efficiency in generating simulations that are representative of a target distribution using an accept-reject mechanism. In our work, we will focus on this approach to efficiently explore and simulate spatial distributions.

3.4.1 Metropolis-Hastings Algorithm

The Metropolis algorithm, introduced by Metropolis et al. in 1953 [55] and generalized by Hastings in 1970 [9], was adapted to spatial processes by Geyer and Møller in 1994 [65]. This algorithm relies on the principle of accepting/rejecting candidates, similar to rejection sampling (see Algorithm 3), and uses a Markov chain to generate samples. The main idea is to propose a new state by perturbing the current state and either accept or reject this proposal. The acceptance rate $R(x, y)$ represents the probability of transitioning from the current state x to the state y. Geyer and Møller's algorithm constructs a Markov chain that explores all possible configurations of the space Ω, with perturbations such as adding or removing an element from the current configuration. In general, the algorithm uses a transition P that is π-reversible and π-invariant. The construction of P takes place in two steps:

- **Proposal transition:** Propose a change $x \to y$ with probability $Q(x, dy)$.
- **Acceptance probability of the change:** Accept this change with probability $a(x, y)$, where $a : \Omega \times \Omega \to]0, 1]$. If the change is rejected, the state remains x.

The two parameters of the algorithm are Q, the proposal transition, and a, the probability of accepting the change. If $\delta_x(dy)$ is the Dirac measure at x, the Metropolis-Hastings transition P is expressed as follows:

$$P(x,dy) = a(x,y)Q(x,dy) + \delta_x(dy)\left[1 - \int_\Omega a(x,z)Q(x,dz)\right]$$

The condition for P to be π-reversible imposes a constraint on Q and the acceptance probability a. Note that $\nu(dx,dy) = \mu(dx,dy) + \mu(dy,dx)$, where $\mu(dx,dy) = Q(x,dy)\pi(dx)$, and μ has a density $h(x,y)$ relative to ν. The Metropolis-Hastings ratio r is defined as:

$$r(x,y) = \frac{h(x,y)}{h(y,x)},$$

for $(x,y) \in R = \{(x,y) \mid h(x,y) > 0 \text{ and } h(y,x) > 0\}$.

The transition P is π-reversible if and only if the acceptance probability a satisfies [66]:

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

If Q and π are respectively of density q and π, and if $q(x,y) > 0$ implies $q(x,y) > 0$, then the π-reversibility of P is written as:

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

Finally, if Q is symmetric, the algorithm simplifies as follows:

```
Algorithm 5: Metropolis-Hastings Algorithm
  Data: Initial state $x_0 = x$, proposal kernel $Q$
  Procedure
    1. Initialization: Choose a state $y$ according to $Q(x,\cdot)$;
    2. Choose a proposal kernel $Q_m$ with probability $p_m$;
    3. Generate $y \sim Q(x,\cdot)$;
    4. Compute the acceptance ratio $a(x,y) = \frac{\pi(y)}{\pi(x)}$;
    5. Accept the state $y$ with probability $a = \min(1, a(x,y))$;
    6. Repeat with the current state;
  Result: Markov chain $x_{0:N}$
```

The Metropolis-Hastings algorithm, while a fundamental method in Markov Chain Monte Carlo (MCMC) algorithms, has several variants adapted to specific contexts. Among these variants are the Metropolis-Hastings random walk algorithm and the

Gibbs sampler, as well as improved versions like the Hastings algorithm and adaptations using more sophisticated proposals to refine performance. Each variant modifies the acceptance mechanism or the choice of proposals to optimize the efficiency and accuracy of simulations. For a detailed exploration of these variants and their applications, we refer the reader to the reference [56].

3.4.2 Metropolis-Hastings Random Walk Algorithm

A particular case of the Metropolis-Hastings algorithm is the random walk case. In this scenario, the proposal is generated by perturbing the current state x, that is, by adding a step ϵ drawn from a density q independent of the current state. We can then write $y = x + \epsilon$, where $\epsilon \sim q$, and the distribution of y given x takes the following form:

$$q(y, x) = q(\epsilon) = q(y - x)$$

The new state, chosen between y and x according to the Metropolis-Hastings acceptance probability, follows the density:

$$p(y, x) = q(y - x)a(y, x) + \left[1 - \int_{\Omega} a(x, z)Q(x, dz)\right] \delta_x(y)$$

thus defining a homogeneous random walk.

Definition 3.15 (Metropolis-Hastings Random Walk Kernel)**.** *Let P be a Metropolis-Hastings kernel with instrumental density q. If the instrumental density is a random walk, i.e., $q(y, x) = q(y - x)$, then the algorithm is said to be of the random walk type.*

3.4.3 Gibbs Sampler

The Gibbs sampler [67] is a particular case of the Metropolis-Hastings algorithm. The main difference between the two algorithms lies in the way new states are generated. While the Metropolis-Hastings algorithm generates new states according to an auxiliary density that is uniform over the set of states, the Gibbs sampler accepts all new states, generating them directly according to the target density towards which we wish to converge. The Gibbs algorithm randomly selects a site i and modifies the value x_i by proposing y_i according to a transition density $q_i(x, y_i)$. The acceptance probability for the Metropolis dynamics is then given by:

$$a_i(x, y) = \min\left(1, \frac{\pi(y) \cdot q_i(y, x_i)}{\pi(x) \cdot q_i(x, y_i)}\right).$$

And the transition kernel is given by:

$$P(x,B) = \sum_i \int_B a_i(x,y) q_i(x,dy) + \left\{ \sum_i \int_\Omega \left[1 - a_i(x,y)\right] q_i(x,dy) - \sum_i q_i(x,\Omega) \right\} \delta_{x \in B}$$

3.5 Conclusion

In conclusion, Markov Chain Monte Carlo (MCMC) methods are powerful tools for complex simulations. The Metropolis-Hastings algorithm excels in generating samples from intricate target distributions through a Markov chain and an acceptance/rejection mechanism.

In the following chapters, we will apply MCMC methods to simulate point processes, a critical step for creating optimized Computer experiment designs. Leveraging the Metropolis-Hastings algorithm will enable tailored and precise analyses across diverse scenarios.

Chapter 4

Construction of Computer Experiment Design from Marked Point Processes

This work constitutes an extension of the study described in [68, 69], with a specific focus on two-marked point processes. We explored the use of marked point processes to simulate the experiments that make up the proposed experiment designs. Unlike classical point processes, two-marked point processes allow the incorporation of geometric and prior knowledge about the points. In particular, Strauss marked processes [6], which account for interactions between pairs of points, are used for this simulation.

To generate these experimental designs, we apply Monte Carlo simulation techniques via Markov chains, more specifically the Metropolis-Hastings algorithm [8, 9]. The experimental points must be optimally distributed within the experimental domain to identify potential irregularities. We also aim to obtain a design where the points are as uniformly distributed as possible within the unit hypercube. The chapter presents a detailed demonstration of the convergence of the Markov chain and provides a comparison of our approach with other existing digital designs, conducted using a program developed in Python, as outlined in the appendix.

All the results presented in this chapter are based on studies conducted in [70, 71] and [72, 73].

4.1 Computer Experiment Design Using Marked Markovian Strauss Point Processes

The main idea is to consider each experiment x_i as a point or particle defined within $[0,1]^p$, and each configuration x as a matrix of experiments. Each point in this configuration is characterized by two marks m_i and m'_i defined in the mark space M. The point and its marks form an object defined as (x_i, m_i, m'_i). Therefore, we equate the objects (experiment design) to realizations of the two-marked point process X. The marked process implies the possibility of interaction. These interactions correspond to neighborhood properties defined in the Ripley-Kelly [46] Markov field. The most commonly used interaction potential is the interaction between pairs of objects. These object processes are crucial for modeling repulsive phenomena. The probability density of a two-marked point process for a configuration x of points is given by:

$$\pi(x) = \alpha \beta_1^{m_1(x)} \beta_2^{m_2(x)} \gamma_{11}^{m_{11}(x)} \gamma_{12}^{m_{12}(x)} \gamma_{22}^{m_{22}(x)} \tag{4.1}$$

Where,

- α is the normalization constant,
- $0 < \gamma_{kl} \leq 1$, where $k \in \{1,2\}$ and $l \in \{1,2\}$ are interaction coefficients,
- β_k, where $k \in \{1,2\}$, is the intensity of the process,
- $m_k(x)$ is the number of points with mark k in x,
- $m_{kl}(x)$ is the number of pairs of $\sim_x$-neighbors of type (k,l) or (l,k) in x (both marked as k and l simultaneously).

4.1.1 Mark Selection

In this study, we characterize the points using two marks: the first one will be the value of the prediction error $\hat{y}_{x_i}$ at the point x_i. Recall that this value is defined as [68,69]:

$$\text{var}(\hat{y}_{x_i}) = {}^t f(x_i) \left({}^t XX\right)^{-1} f(x_i)$$

Where,

- $X = {}^t[f(x_1), f(x_2), \ldots, f(x_n)]$ is the computation matrix, which depends on the chosen experimental points and the assumed model,

- $\left({}^{t}XX\right)^{-1}$ is the dispersion matrix,
- $f\left(x_i\right)$ is the modeled vector for point x_i.

In this case, we define $n_1\left(x\right)$ for a configuration x as follows:

$$m_1(x) = \sum_{i=1}^{n} 1_{\text{var}\left(\hat{y}_{x_i}\right)\leq\varepsilon}$$

As a second mark, we will take the average of the normal density distances between the point x_i and the other points in the configuration x. This mark will be given by:

$$m_2(x) = \sum_{i=1}^{n} 1_{\mu(x_i)\geq r}$$

Where $\mu\left(x_i\right) = \frac{1}{n-1}\sum\limits_{\substack{i=1\\ j\neq i}}^{n} \delta\left(x_i, x_j\right)$ with $\delta\left(x_i, x_j\right) = \int\limits_0^l \varphi(t)dt$, where l is the usual distance between points x_i and x_j. φ represents the density of the normal distribution where ε and r are fixed values.

4.2 Simulation of Point Processes using the MCMC Method and the Metropolis-Hastings Algorithm

This method involves constructing a chain $\{X_0, X_1, \ldots, X_N\}$ that converges to the desired distribution π. In fact, the Metropolis-Hastings (MH) algorithm can perform this construction using the π-reversible transition kernel. Recall that the algorithm goes through two steps.

- We propose a state change from x to y according to the probability distribution $Q\left(x, \cdot\right)$,
- We accept y with probability $a(x, y)$, otherwise, we stay in the state x (Where $a : \Omega \times \Omega \mapsto [0, 1]$).

Let $q(x, y)$ be the density of $Q\left(x, \cdot\right)$, the MH transition is written as [8]:

$$P_{MH}\left(x, y\right) = a\left(x, y\right) q\left(x, y\right) + \left[1 - \int\limits_{\Omega} a\left(x, z\right) q\left(x, z\right) dz\right] \delta_x(y)$$

With $\delta_x(.)$ representing the point mass at x. To simplify calculations, we use the Dirac measure at x ($\delta_x\,(y) = 1$ if $x = y$ and 0 otherwise).

The choice of (Q, a) will ensure the π-reversibility of P_{MH} if the following equilibrium equation is satisfied:

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

The choice of the acceptance probability $a(x, y)$ is more constrained: it is essentially dictated by the goal of (asymptotically) simulating a given probability distribution π. This is the case in the usual choice, where:

$$a\,(x,\; y) \;=\; \frac{\pi\,(y) \times q\,(y,\; x)}{\pi\,(x) \times q\,(x,\; y)}$$

Two important points to note. Firstly, the calculation of $a\,(x,\; y)$ does not require any knowledge of the normalization constant in (4.1). Secondly, in this work, we consider the case where two configurations x and y differ in exactly one point. This is referred to as the 'spin flop dynamics,' and thus, the density q is symmetric: $q\,(y, x) = q\,(x, y)$ In this case, the acceptance probability reduces to:

$$a(x, y) = \frac{\pi\,(y)}{\pi(x)} = \frac{\beta_1^{m_1(y)}\beta_2^{m_2(y)}\gamma_{11}^{m_{11}(y)}\gamma_{12}^{m_{12}(y)}\gamma_{22}^{m_{22}(y)}}{\beta_1^{m_1(x)}\beta_2^{m_2(x)}\gamma_{11}^{m_{11}(x)}\gamma_{12}^{m_{12}(x)}\gamma_{22}^{m_{22}(x)}}$$

4.2.1 The algorithm for constructing the proposed experiment design

The computer experiment design proposed in this work (referred to as the two-type marked experiment design) is generated using the following algorithm 6:

Algorithm 6: Proposed Design Construction Algorithm

```
• Initialization Step: Choose an initial configuration (experimental design)
  $\left(X_0 = x \text{ or } x = (x_1, x_2, \ldots, x_n) \text{ and } x \in [0,1]^k\right)$ according to a given probability
  distribution, for example, the uniform distribution.

• Iteration Step:

for $N = 1, 2, ..., N_{MCMC}$ do
    for For each state $x$ do
        Sample $y$ using the spin-flip dynamics.
            ◦ Randomly choose a spin $s$ uniformly from $\{1, \cdots, n\}$.
            ◦ Simulate an experiment $y_j$ according to the uniform distribution on $[0,1]^p$.
              Then take the new configuration as: $y = \left(x_1, x_2, \cdots x_{s-1}, y_j, x_{s+1} \cdots, x_n\right)$.
    end
        • Calculate the acceptance probability:
          $a(x,y) = \min\left(1; \beta_1^{m_1(y)-m_1(x)} \beta_2^{m_2(y)-m_2(x)} \gamma_{11}^{m_{11}(y)-m_{11}(x)} \gamma_{12}^{m_{12}(y)-m_{12}(x)} \gamma_{22}^{m_{22}(y)-m_{22}(x)}\right)$
        • Choose $x = \begin{cases} y \text{ with probability } a \\ x \text{ with probability } 1-a \end{cases}$.
    Repeat the last two steps $n$ times for each iteration $N$.
    Take $X_N = x$
end
```

For $N = 1000$, Figure 4.1 shows the convergence towards a configuration that characterizes the realization of a two-marked Strauss point process starting from an initial configuration of 35 points uniformly chosen in $[0,1]^2$:

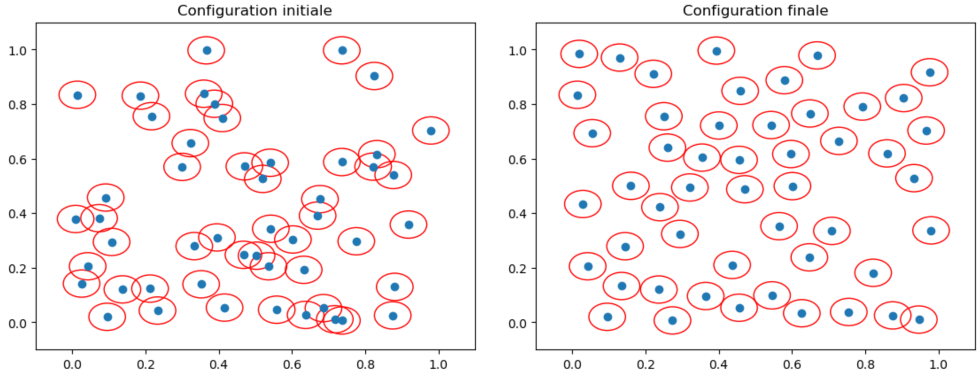

Figure 4.1: On the left, an initial configuration of 35 points, and on the right, a final configuration for $\gamma_{11} = 0.01$, $\gamma_{12} = 0.01$, $\gamma_{22} = 0.05$, $\beta_1 = 0.9$, $\beta_2 = 1.5$, and $r = 0.1$.

The interactions between the experiments will be represented in Figure 1 by drawing circles with a radius of ${}^{r}/_{2}$. When two circles intersect, it indicates a specific interaction between those experiments. If the radius r is too small, the distribution shows no interactions. Conversely, if the radius is too large, the distribution exhibits clusters. Therefore, it is important to choose an appropriate radius to avoid such issues.

4.3 Convergence study

For each iteration N of the construction algorithm described above, we perform n basic transformations. Therefore, the chain of experimental designs $(X_N)_{N\geq 0}$ generated in this way is the realization of a Markov chain with the transition kernel:

$$P\left(x,y\right) = P_{MH}^{n}\left(x,y\right)$$

At this point, the fundamental question is whether the chain converges to the distribution $\pi(x)$ defined in (4.1). The chain converges to the invariant distribution π if:

$$P^{t}\left(x,A\right) \underset{t\to\infty}{\rightarrow} \pi\left(A\right)$$

Where A is a Borel set from $\mathcal{A}$, and $P^{t}\left(x,A\right) = p(X_t = A/X_0 = x)$ is a transition kernel at time t. Let's state the main result of interest here:

Proposition 4.1. *On a finite space, the transition kernel P of the Markov chain $(X_N)_{N\geq 0}$ obtained from the construction algorithm is positive recurrent, π-stationary, aperiodic, and primitive (primitive kernel).*

Proof. First, we demonstrate three important properties for the kernel P_{MH}: π-reversibility, π-stationarity, and π-irreducibility.

- π**-reversibility:** By definition, the transition P_{MH} is π-reversible if:

$$\forall x,\ y\ \in\ \Omega\ :\ \pi\left(x\right)P_{MH}\left(x,\ y\right)\ =\ \pi\left(y\right)P_{MH}\left(y,\ x\right)$$

Let $x \in \Omega$ and $B \in \mathcal{A}$, we have:

$$\int\limits_{\Omega} 1_{B(x,y)}\pi\left(x\right)P_{MH}\left(x,y\right)dx = \int\limits_{\Omega} 1_{B(x,y)}\pi\left(x\right)a\left(x,y\right)q\left(x,y\right)dx$$

$$+\int\limits_{\Omega} 1_{B(x,y)}\pi\left(x\right)\left[1-\int\limits_{\Omega} a\left(x,z\right)q\left(x,z\right)dz\right]\delta_x\left(y\right)dx$$

$$=\int\limits_{\Omega} 1_{B(x,y)}\pi\left(x\right)a\left(x,y\right)q\left(x,y\right)dx$$

$$+\int\limits_{\Omega} 1_{B(x,y)}\delta_x\left(y\right)\pi\left(x\right)\left[1-\int\limits_{\Omega} a\left(x,z\right)q\left(x,z\right)dz\right]dx$$

$$=\int\limits_{\Omega} 1_{B(x,y)}\pi\left(x\right)a\left(x,y\right)q\left(x,y\right)dx$$

$$+\int\limits_{\Omega} 1_{B(x,x)}\pi\left(x\right)\left[1-\int\limits_{\Omega} a\left(x,z\right)q\left(x,z\right)dz\right]dx$$

And since:

$$\pi\left(x\right)a\left(x,y\right)q\left(x,y\right) = \alpha\beta_1^{m_1(x)}\beta_2^{m_2(x)}\gamma_{11}^{m_{11}(x)}\gamma_{12}^{m_{12}(x)}\gamma_{22}^{m_{22}(x)}$$

$$\times\min\left(1;\beta_1^{m_1(y)-m_1(x)}\beta_2^{m_2(y)-m_2(x)}\gamma_{11}^{m_{11}(y)-m_{11}(x)}\gamma_{12}^{m_{12}(y)-m_{12}(x)}\gamma_{22}^{m_{22}(y)-m_{22}(x)}\right)q(x,y)$$

$$=\alpha\min\left(\beta_1^{m_1(x)}\beta_2^{m_2(x)}\gamma_{11}^{m_{11}(x)}\gamma_{12}^{m_{12}(x)}\gamma_{22}^{m_{22}(x)},\beta_1^{m_1(y)}\beta_2^{m_2(y)}\gamma_{11}^{m_{11}(y)}\gamma_{12}^{m_{12}(y)}\gamma_{22}^{m_{22}(y)}\right)q(x,y)$$

$$=\alpha\beta_1^{m_1(y)}\beta_2^{m_2(y)}\gamma_{11}^{m_{11}(y)}\gamma_{12}^{m_{12}(y)}\gamma_{22}^{m_{22}(y)}$$

$$\times\min\left(\beta_1^{m_1(x)-m_1(y)}\beta_2^{m_2(x)-m_2(y)}\gamma_{11}^{m_{11}(x)-m_{11}(y)}\gamma_{12}^{m_{12}(x)-m_{12}(y)}\gamma_{22}^{m_{22}(x)-m_{22}(y)};1\right)\times q(x,y)$$

$$=\pi(y)\min\left(1;\beta_1^{m_1(x)-m_1(y)}\beta_2^{m_2(x)-m_2(y)}\gamma_{11}^{m_{11}(x)-m_{11}(y)}\gamma_{12}^{m_{12}(x)-m_{12}(y)}\gamma_{22}^{m_{22}(x)-m_{22}(y)}\right)q(x,y)$$

$$=\pi(y)a(y,x)q(x,y)$$

And since $q\left(x,y\right)=q\left(y,x\right)$, then:

$$\pi\left(x\right)a\left(x,\ y\right)q\left(x,y\right)=\pi\left(y\right)a\left(y,x\right)q\left(y,x\right)$$

, We obtain:

$$\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)P_{MH}\left(x,y\right)dx$$

$$=\int\limits_{\Omega}1_{B(x,y)}\pi\left(y\right)a\left(y,x\right)q\left(y,x\right)dx+\int\limits_{\Omega}1_{B(y,y)}\pi\left(y\right)\left[1-\int\limits_{\Omega}a\left(y,z\right)q\left(y,z\right)dz\right]dy$$

$$=\int\limits_{\Omega}1_{B(x,y)}\pi\left(y\right)P_{MH}\left(y,x\right)dy$$

So $\pi\left(x\right)P_{MH}\left(x,\ y\right)\ =\ \pi\left(y\right)P_{MH}\left(y,\ x\right)$, and therefore, the chain is π-reversible.

- **π-stationarity:** The transition P_{MH} is π-stationary if:

$$\forall x,y\ \in\Omega;A,B\in\mathcal{A}:\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)P_{MH}\left(x,A\right)dx=\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)dx$$

Let $x\in\Omega$ and $B\in\mathcal{A}$. We then have:

$$\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)P_{MH}\left(x,y\right)dx=\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)\left[\int\limits_{\Omega}a\left(x,y\right)q\left(x,y\right)dy\right]dx$$

$$+\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)\left[\int\limits_{\Omega}1-a\left(x,z\right)q\left(x,z\right)dz\right]\delta_{x}\left(y\right)dx$$

$$=\int\limits_{\Omega}\int\limits_{\Omega}1_{B(x,y)}\pi\left(x\right)a\left(x,y\right)q\left(x,y\right)dydx+\int\limits_{\Omega}1_{B(x,x)}\pi\left(x\right)dx-$$

$$\int\limits_{\Omega}\int\limits_{\Omega}\pi\left(x\right)a\left(x,z\right)q\left(x,z\right)dzdx=\int\limits_{\Omega}1_{B(x,x)}\pi\left(x\right)dx$$

So the chain admits π as a stationary distribution.

- **π-irreducibility:** The transition P_{MH} is π-irreducible if:

$$\forall A\in\mathcal{A},\pi\left(A\right)>0\ \Rightarrow\ \exists t\ ,\ \ P_{MH}^{t}\left(x,A\right)>0$$

Let A be a Borel set from $\mathcal{A}$, and for $t = 1$ we have:

$$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}\left(x, A\right) dx = \int\limits_{\Omega} 1_{B(x,A)} a\left(x, A\right) q\left(x, A\right) dx$$

$$+ \int\limits_{\Omega} 1_{B(x,A)} \left[1 - \int\limits_{\Omega} a\left(x, z\right) q\left(x, z\right) dz\right] \delta_x\left(A\right) dx$$

$$= \int\limits_{\Omega} 1_{B(x,A)} a\left(x, A\right) q\left(x, A\right) dx + \int\limits_{\Omega} 1_{B(x,x)} \left[1 - \int\limits_{\Omega} a\left(x, z\right) q\left(x, z\right) dz\right] dx$$

$$= \int\limits_{\Omega} 1_{B(x,A)} a\left(x, A\right) q\left(x, A\right) dx + 1 - \int\limits_{\Omega}\int\limits_{\Omega} a\left(x, z\right) q\left(x, z\right) dzdx$$

Since:

$a(x, A) = \min\left(1; \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)}\right)$

and

$a(x, z) = \min\left(1; \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)} \gamma_{22}^{m_{22}(z)-m_{22}(x)}\right)$

Then we have four possible cases:

- if $a\left(x, A\right) = 1$ and

  $$a(x, z) = \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)} \gamma_{22}^{m_{22}(z)-m_{22}(x)}$$

  then:

  $$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}\left(x, A\right) dx = \int\limits_{\Omega} 1_{B(x,A)} q\left(x, A\right) dx + 1$$

  $$- \int\limits_{\Omega}\int\limits_{\Omega} \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)} \times$$

  $$\gamma_{22}^{m_{22}(z)-m_{22}(x)} q\left(x, z\right) dzdx = \int\limits_{\Omega} 1_{B(x,A)} q\left(x, A\right) dx + 1$$

  $$- \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)} \gamma_{22}^{m_{22}(z)-m_{12}(x)} > 0$$

- if $a(x, z) = \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)}$ and

$a\left(x,z\right)=1$ then:

$$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}\left(x,A\right)dx$$

$$= \int\limits_{\Omega} 1_{B(x,A)} \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \times$$

$$\gamma_{22}^{m_{22}(A)-m_{22}(x)} q\left(x,A\right)dx + 1\ - \int\limits_{\Omega}\int\limits_{\Omega} q\left(x,z\right)dzdx$$

$$= \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)} \times$$

$$\int\limits_{\Omega} 1_{B(x,A)} q\left(x,A\right)dx > 0$$

- ○ if $a(x,z) = \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)}$ and $a(x,z) = \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)} \gamma_{22}^{m_{22}(z)-m_{22}(x)}$ then:

$$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}\left(x,A\right)dx$$

$$= \int\limits_{\Omega} 1_{B(x,A)} \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma$$

$$_{22}^{m_{22}(A)-m_{22}(x)} q\left(x,A\right)dx$$

$$+1 - \int\limits_{\Omega}\int\limits_{\Omega} \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \gamma_{12}^{m_{12}(z)-m_{12}(x)}$$

$$\times \gamma_{22}^{m_{22}(z)-m_{22}(x)} q\left(x,z\right)dzdx$$

$$= \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)} \times$$

$$\int\limits_{\Omega} 1_{B(x,A)} q\left(x,A\right)dx + 1 - \beta_1^{m_1(z)-m_1(x)} \beta_2^{m_2(z)-m_2(x)} \gamma_{11}^{m_{11}(z)-m_{11}(x)} \times$$

$$\gamma_{12}^{m_{12}(z)-m_{12}(x)} \gamma_{22}^{m_{22}(z)-m_{22}(x)} \int\limits_{\Omega}\int\limits_{\Omega} q\left(x,z\right)dzdx$$

$$= \beta_1^{m_1(A)-m_1(x)} \beta_2^{m_2(A)-m_2(x)} \gamma_{11}^{m_{11}(A)-m_{11}(x)} \gamma_{12}^{m_{12}(A)-m_{12}(x)} \gamma_{22}^{m_{22}(A)-m_{22}(x)} \times$$

$$\int\limits_{\Omega} 1_{B(x,A)} q\left(x,A\right)dx + 1$$

$$-\beta_1^{m_1(z)-m_1(x)}\beta_2^{m_2(z)-m_2(x)}\gamma_{11}^{m_{11}(z)-m_{11}(x)}\gamma_{12}^{m_{12}(z)-m_{12}(x)}\gamma_{22}^{m_{22}(z)-m_{22}(x)} > 0$$

So $\int_\Omega 1_{B(x,A)} {P_{MH}}^t\,(x,A)\,dx > 0 \;\; \forall\, t \geq 0$, then P_{MH}est π-irreducible.

Since π is the invariant distribution of P_{MH}, it is also an invariant distribution for P. Indeed, $\pi P_{MH} = \pi$, and by induction on the integer, $\pi P_{MH} = \pi$, we obtain:

$$\pi P_{MH} = \pi P_{MH}^2 = \pi P_{MH}^3 = \ldots = \pi P_{MH}^n = \pi$$

So, $\pi P = \pi$. By construction of $P = P_{MH}^n$, the π-irreducibility of P_{MH} implies the π-irreducibility of P. If P is π-irreducible and has an invariant distribution π, then P is positive recurrent, and π is the unique invariant distribution of P. By construction of $P = P_{MH}^n$, we have $\pi P = \pi$. If P is π-irreducible and has an invariant distribution π, then P is positive recurrent, and π is the unique invariant distribution of P [62] (see proposition 1).

Furthermore, the chain created by the construction algorithm will also be aperiodic as long as there exists at least one pair of configurations (x, y) such that $a\,(x, y) < 1$, because then we have $P\,(x, x) > 0$. It is quickly evident that the chain is aperiodic, as the event $X_{(N+1)} = X_{(N)}$ is possible practically at any time. Indeed, each state can be visited in two consecutive iterations, so $P^1\,(x, x) > 0$, making their period 1.

Since the chain generated by the algorithm is irreducible and aperiodic, its transition kernel P is primitive (a characterization of a primitive Markov Kernel more common in probability theory is to say that it is irreducible and aperiodic). □

Theorem 4.1. *The Markov chain* $(X_N)_{N \geq 0}$ *obtained from the proposed construction algorithm is geometrically ergodic, and its kernel P simulates a marked point process with two types of density:*

$$\pi\,(x) = \alpha\beta_1^{m_1(x)}\beta_2^{m_2(x)}\gamma_{11}^{m_{11}(x)}\gamma_{12}^{m_{12}(x)}\gamma_{22}^{m_{22}(x)}$$

In other words, vP^m *converges to* π *as m tends to infinity, where v is the initial distribution, and we have:*

$$\lim_{m\to\infty} \|vP^m - \pi\| = 0$$

Proof. Let v be an initial distribution, for any integer m and for all $x \in N^{lf}$, we have:

$$\|vP^m\,(x, .) - \pi\| = \|vP^m - \pi P^m\| \leq 2C\,(P^m) \leq 2(C\,(P))^m$$

Where $C\,(P)$ is the Dobrushin contraction coefficient of P [63].

Where $C(P)$ is the Dobrushin contraction coefficient of P [64]. According to Proposition 4.1, the kernel P is primitive, thus $0 \leq C(P) < 1$ (see Lemma 3.2). Hence, as m tends to infinity, $\|vP^m - \pi\| \underset{m\to\infty}{\rightarrow} 0$. Therefore, the chain is uniformly ergodic and converges to the distribution defined in (4.1).

□

4.4 Numerical Results and Quality of the Proposed Designs

The evaluation of the quality of the proposed Computer experiment design relies on the use of common criteria aimed at ensuring an optimal filling of the experimental space and a uniform distribution of points. This evaluation is crucial in the experiment design process, as a well-designed plan maximizes the efficiency of experiments and provides meaningful and reliable results.

In our evaluation of the Computer experiment designs, we use three types of criteria:

- **Distance criterion:** aims to maximize the minimum distance between two points in the design. A high value of this criterion indicates a maximal dispersion of points.

- **Coverage criterion:** evaluates the deviation between the points of the design and those of a regular grid, with a zero value for a regular grid. Minimizing coverage brings the design closer to a regular distribution and ensures adequate filling of the space.

- **Discrepancy criterion:** measures the deviation between the empirical distribution function of the points in the design and that of the uniform distribution. A low discrepancy, particularly in the L_2 norm, indicates a more uniform distribution of points.

Table 1 provides a comparison based on the discrepancy criterion between the designs proposed in this study, referred to as TMD (Two Mark Designs), and low discrepancy sequences such as the Halton sequence [32], Sobol sequence [33], and Faure sequence [34]. It is noteworthy that the proposed designs exhibit low discrepancy, comparable to that of the mentioned low discrepancy sequences. This observation highlights the quality of the proposed experiment designs, demonstrating their ability to provide a uniform distribution of points in the experimental space, while competing with well-established methods of generating low discrepancy sequences.

Tableau 4.1: Discrepancy values for the proposed designs (TMD), the Halton sequence, Sobol sequence, and Faure sequence for 4, 7, and 10 factors.

Number of Factors	Number of Points	TMD	Halton Sequence	Sobol Sequence	Faure Sequence
4	32	0.00176	0.001779	0.000843	0.001641
7	64	0.0001207	0.00048	0.000224	0.000480
10	128	0.00000696	0.000109	0.0000605	0.000109

In this chapter, a comparison is made between the constructed designs and the stochastic designs commonly used in Computer experiments, excluding low discrepancy sequences. To give appropriate meaning to the results obtained, the criteria were calculated on a set of 100 designs. The designs compared in this section include:

- Random designs (RD)
- Latin Hypercubes (LHS) [31]
- Maximin LHS designs (mLHS) [74]
- Strauss designs (SD) [75]
- Maximal entropy designs (Dmax) [76]
- Marked Strauss designs [68, 69]
- Connected component model designs (CCD) [77]
- Proposed designs (TMD).

Figures 4.2 and 4.3 present the results of the various criteria in the form of box plots for 5 and 7 dimensions.

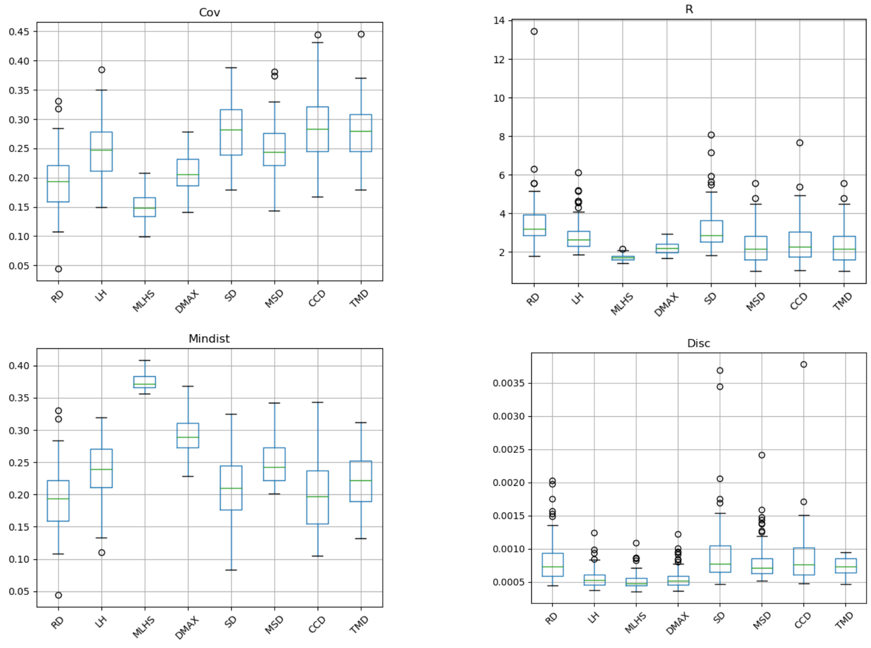

Figure 4.2: Box plots of the quality criteria calculated on 100 designs with 30 points in 5 dimensions.

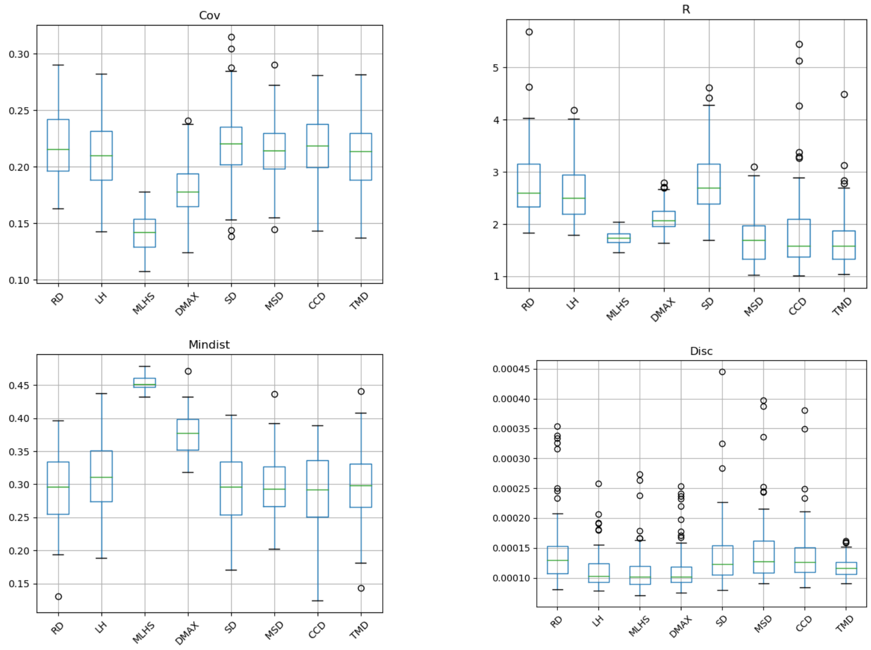

Figure 4.3: Box plots of quality criteria calculated for 80 designs with 50 points in 7 dimensions.

A few remarks about the figures above are worth noting. The maximal entropy designs, Latin Hypercubes, maximin LHS designs, connected component models, and two-mark designs all performed well with respect to the discrepancy criterion. It is interesting to note that the two-mark designs are among the aforementioned designs that also achieved very good results according to the R criterion.

4.5 Conclusion

In this chapter, we have presented our theoretical contribution to the methodology of experimental research, with a focus on the innovative application of the Markov Chain Monte Carlo (MCMC) method. Specifically, we explored the integration of Strauss marked point processes to design new computer experiment plans, a methodology that stands out for its ability to incorporate geometric knowledge and prior information into experiment design.

This approach proves to be an interesting alternative to traditional statistical methods, opening up new perspectives for more precise research that is tailored to complex experiment contexts. By comparing our method with other computer approaches using various evaluation criteria, we found that our proposal not only delivers satisfactory results but also shows promise for creating experiment designs in demanding, multifactorial environments.

Thus, this work contributes to enriching the methodological tools available to researchers by offering an innovative solution that addresses the contemporary challenges of experimental research.

Chapter 5

New Computer Experiment Designs Using Point Processes with Area Interaction

This chapter presents an innovative method for constructing computer experiment designs based on the theory of point processes with area interaction [7]. This method is essential for understanding the interactions between different elements within a modeled system, offering a more flexible and adaptable approach compared to traditional mathematical models. Unlike conventional approximate models that rely on simplified equations, our method uses the Markov Chain Monte Carlo (MCMC) method and the Metropolis-Hastings algorithm combined with Voronoi tessellations. It employs a new dynamic called homogeneous birth-and-death dynamics of a set of points to generate the designs. This approach does not require the development of specific mathematical models for each system studied, making it universally applicable while achieving comparable results. Additionally, we provide an in-depth analysis of the Markov chain's convergence properties to ensure the generated designs' reliability. Our approach and other existing computer experiment designs are also compared.

This chapter presents results from investigations carried out in [78,79] and [80,81].

5.1 Computer Experiments Designs Using Markovian Area-Interaction Point Processes

Each experiment x_i is considered as a point or particle defined on $[0,1]^p$, and each configuration x is regarded as a computer experiment design. Hence, the n experiments can be seen as realizations of an area-interaction point process [7]. This interaction corresponds to neighborhood properties, as defined by a Markov field in the sense of Ripley–Kelly [46]. This process is essential for modeling repulsive phenomena. The density of the area-interaction point process is defined by the following:

$$\pi\left(x\right) = \alpha\beta^{n(x)}\gamma^{-m(U_r(x))} \tag{5.1}$$

where $\alpha > 0$ is the normalization constant that makes π a density, $n(x)$ is the number of points in the configuration x, $\beta > 0$ is a scale parameter, $\gamma > 0$ is a repulsion parameter, $m(\cdot)$ is the Lebesgue measure, $r > 0$ is the radius of the ball $B\left(x_i, r\right)$ is defined by $B\left(x_i, r\right) = \left\{a \in [0,1]^p, \|x_i - a\| \leq r\right\}$, and $U_r(x) = \bigcup\limits_{i=1}^{n} B\left(x_i, r\right)$ is the union of balls centered at x_i with radius r [7,82,83].

The area of the union of discs may be expressed as the decomposition of the union of grains, $U_r(x) = \bigcup\limits_{i=1}^{n} B\left(x_i, r\right)$, in an inclusion–exclusion style [5,83,84]. This is expressed concisely as follows:

$$m\left(U_r(x)\right) = \sum_{i=1}^{n(x)} m\left(B\left(x_i, r\right)\right) - \sum_{i<j} m\left(B\left(x_i, r\right) \cap B\left(x_j, r\right)\right) + \cdots$$

$$+ (-1)^{n(x)+1} m\left(\bigcap_{i=1}^{n(x)} B\left(x_i, r\right)\right)$$

Figure 5.1 illustrates an example of $U_r(x)$ on the square $[0,1]^2$. The red points x_i and the blue disks of the center x_i and radius r are shown. The blue shaded area in this figure represents $U_r(x)$, and $m(U_r(x))$ denotes the area of this region.

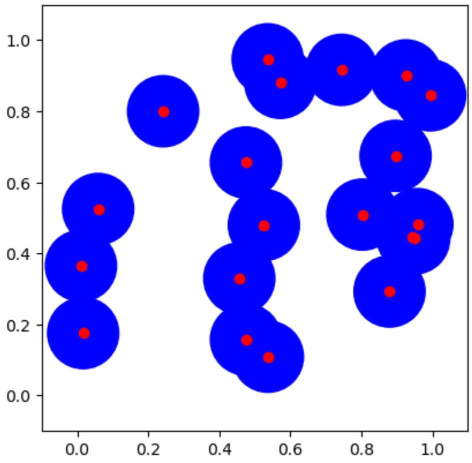

Figure 5.1: Example of $U_r(x)$ for 20 points in the unit square $[0,1]^2$.

Simulation of Point Processes Using the Markov Chain Monte Carlo (MCMC) Method and the Metropolis-Hastings Algorithm

This method involves constructing a chain $\{X_0, X_1, \cdots, X_N\}$ that converges to the desired distribution π. In fact, the Metropolis–Hastings algorithm achieves this construction by using a π-reversible transition kernel. The algorithm goes through two steps:

- We make a proposal for a state change: from x to y, according to a probability law $Q(x,.)$ with density $q_m(x,y)$, called the instrumental density.
- We accept y with probability $a_m(x,y)$; otherwise, we stay in state x (where $a_m : \Omega \times \Omega \to [0,1]$).

The transition kernel is given by [60]:

$$P(x,B) = \sum_m \int_B a_m(x,y)\, q_m(x,y)\, dy + \left\{ \sum_m \int_\Omega [1 - a_m(x,y)]\, q_m(x,y)\, dy - \sum_m q_m(x,\Omega) \right\} \delta_{x\in B}$$

In our case, we take $q_m(x,\ y) = q(x,\ y)$ and $a_m(x,\ y) = a(x,\ y)$ (i.e., a homogeneous Markov chain). From this, we can deduce the Metropolis–Hastings transition matrix P_{MH}, which is given by [8]:

$$P_{MH}(x,y) = a(x,y)\, q(x,y) + \left[\int_\Omega 1 - a(x,z)\, q(x,z)\, dz \right] \delta_x(y)$$

where $\delta_x(\cdot)$ represents the mass at point x; for simplification, we use the Dirac measure at x ($\delta_x\left(y\right) = 1$ if $x = y$, and 0 otherwise).

The choice of (Q, a) will ensure the π-reversibility of P_{MH} if the following equilibrium equation is satisfied:

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

The choice of the acceptance probability a is more limited and essentially dictated by the objective of simulating (asymptotically) a given probability distribution π. This is the case for the usual choice, where:

$$a\left(x,\ y\right)\ = \frac{\pi\left(y\right) \times q\left(y,\ x\right)}{\pi\left(x\right) \times q\left(x, y\right)}$$

It is important to note certain points. Firstly, the calculation of $a(x, y)$ does not require knowing the normalization constant of (5.1). Secondly, in this work, we consider the case where both configurations, x and y, differ at multiple points, which is called the homogeneous birth and death dynamics of a set of points. As a result, the chosen density q is usually symmetric to make the process computationally manageable: $q\left(x, y\right) = q\left(y, x\right)$. Thus, the acceptance probability is reduced to the following:

$$a\left(x,\ y\right) = \frac{\pi\left(y\right)}{\pi(x)} = \frac{\beta^{n(y)}\gamma^{-m(U_r(y))}}{\beta^{n(x)}\gamma^{-m(U_r(x))}} = \frac{\gamma^{-m(U_r(y))}}{\gamma^{-m(U_r(x))}}$$

5.2 Algorithm for Constructing Computer Experiments Designs Using Markovian Area-Interaction Point Processes

The computer experiment designs proposed in this work were generated using Algorithm 7, which is essentially a version of the Metropolis–Hastings algorithm with the use of Voronoi tessellations.

Algorithm 7: Metropolis–Hastings Algorithm with Voronoi Tessellations.

- **Initialization Step:** Choose an initial configuration (an experimental design) $X_0 = x$ according to a given probability distribution, for example, the uniform distribution.

- **Iteration Step:**

```
for $i = 1, 2, \ldots, N_{MCMC}$ do
    for each configuration $x$ do
        - Subdivide the configuration $x$ into neighborhood zones $\vartheta(x_k)$ for
          each point $x_k$, $k \in \{1, 2, \cdots, n\}$ using Voronoi-Dirichlet tessellations.
        - For each neighborhood $\vartheta(x_k)$, simulate an experiment $y_k$ according to
          the proposal distribution $q \sim U_{\vartheta(x_k)}$.
        - Take $y = x_i \cup \{y_1, y_2, \cdots, y_n\}$.
        - Choose a pair of points $\{u, v\}$ from the configuration $y$ randomly, then
          choose either $u$ or $v$ with a probability of $1/2$ and remove it from $y$
          (i.e., $y = \begin{cases} y/\ \{u\} \text{ if } u \text{ is chosen,} \\ y/\ \{v\} \text{ if } v \text{ is chosen.} \end{cases}$)
          Repeat this step $n$ times.
        The new configuration is then taken as $y$.
    end
    Compute the acceptance probability $a(x, y) = \min\left(1, \gamma^{m(U_r(x)) - m(U_r(y))}\right)$.
    Update $x$ as follows:
        $x = \begin{cases} y \text{ with probability } a, \\ x \text{ with probability } 1 - a. \end{cases}$
end
Result: Return $X_N = x$
```

For $N = 1000$, Figures 4.1 and 5.3 show the convergence towards a configuration that characterizes the realization of an area-interaction point process starting from an initial configuration of 25 points, chosen uniformly in $[0, 1]^2$ and $[0, 1]^3$, respectively.

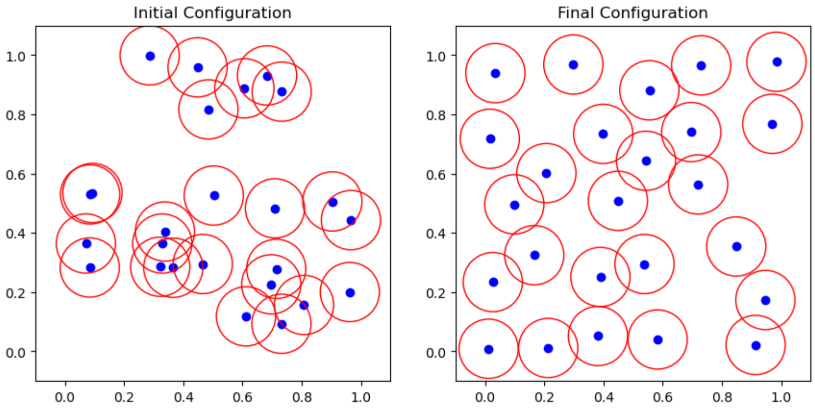

Figure 5.2: On the left, an initial configuration of 25 points with $m(U_r(x)) = 0.5408$, and on the right, a final configuration for $\gamma = 3$ and $r = 0.1$ with $m(U_r(x)) = 0.7594$.

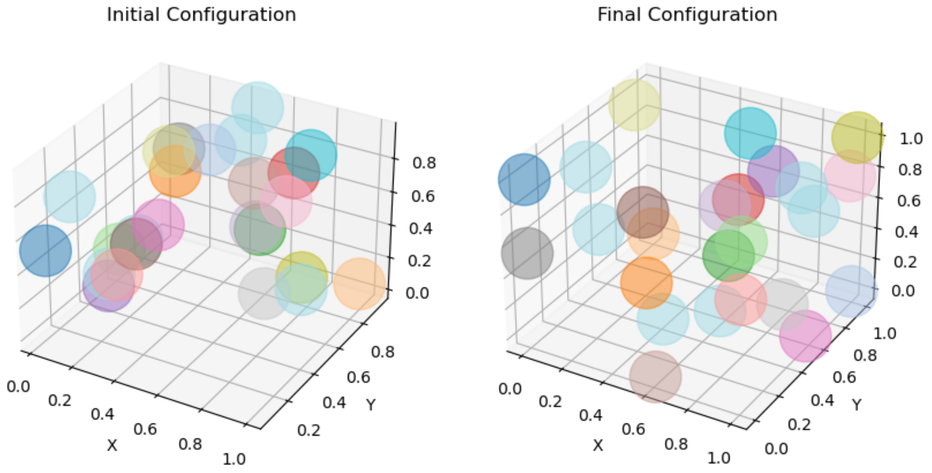

Figure 5.3: On the left, an initial configuration of 25 points with $m(U_r(x)) = 0.1040$, and on the right, a final configuration for $\gamma = 3$ and $r = 0.1$ with $m(U_r(x)) = 0.1047$.

Influence of Parameters

Figure 5.4 shows the impact of the parameter r on the final distribution of points. It is crucial to choose the radius r wisely, as a too-small radius results in a distribution without interaction but with numerous gaps. On the other hand, a too-large radius leads to a distribution with excessive interaction.

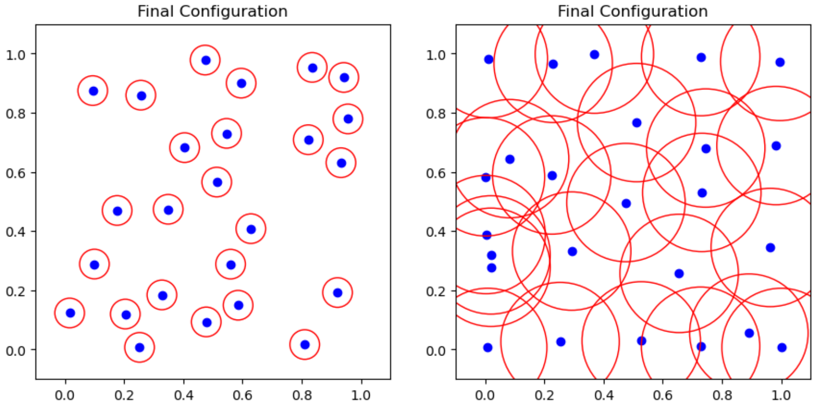

Figure 5.4: On the left, an initial configuration of 25 points with $m\left(U_r(x)\right) = 0.5408$, and on the right, a final configuration for $\gamma = 3$ and $r = 0.1$ with $m\left(U_r(x)\right) = 0.7594$.

The interaction radius is the most sensitive parameter to adjust, and its value must be carefully selected. For a given criterion, the best solution would likely be to tabulate this value according to the number of points and the dimension of the problem.

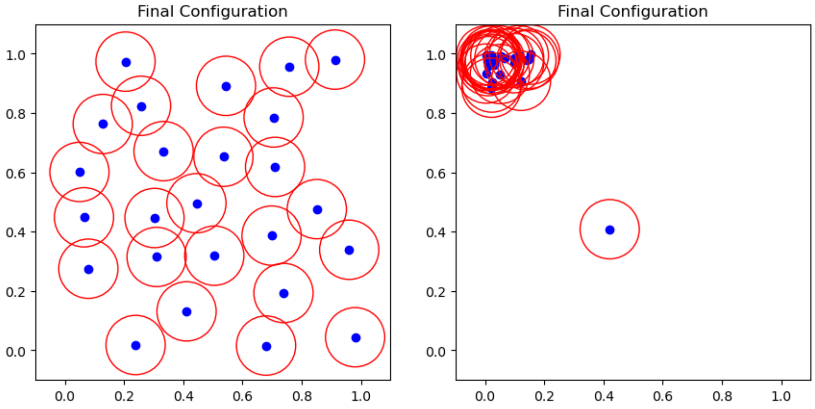

Figure 5.5: On the left, a configuration of 25 points with $\gamma = 3$ and $r = 0.05$ with $m\left(Ur(x)\right) = 0.1960$, and on the right, a configuration of 25 points with $\gamma = 3$ and $r = 0.3$ with $m\left(Ur(x)\right) = 1.7691$.

5.3 Convergence of the Proposed Algorithm

For each iteration N of the algorithm described above, the chain of computer experiments designs $\left(X_N\right)_{N\geq 0}$ generated is a realization of a Markov chain with the transition kernel:

$$P\left(x, y\right) = P_{MH}\left(x, y\right)$$

The essential question that arises is whether the chain converges to the distribution $\pi\left(x\right)$ defined in (5.1).

Definition 5.1. *The chain is convergent to the invariant distribution π if:*

$$\lim_{n\to\infty} P^n\left(x, A\right) = \pi\left(A\right)$$

where A is a Borel set in $\mathcal{A}$, $P^n\left(X_0, A\right) = P\left(X_n \in A | X_0\right)$ with:

$$P\left(X_n, A\right) = P\left(X_{n+1} \in A | X_0, X_1, \ldots, X_n\right) \quad \textit{and} \quad \pi\left(A\right) = \int \pi\left(dx\right) P\left(x, A\right)$$

Proposition 5.1. *On a finite space, the transition kernel P of the Markov chain $\left(X_N\right)_{N\geq 0}$ obtained from the construction algorithm is recurrent positive, π-stationary, aperiodic, and primitive (primitive kernel).*

Proof. Firstly, we show three important properties for the kernel P_{MH}: π-reversibility, π-stationarity, and π-irreducibility.

- **π-reversibility**
  The transition P_{MH} is π-reversible if, for any sets A and B, the probability of transitioning from a state in A to a state in B is equal to the probability of transitioning from a state in B to a state in A ($A, B \subset \Omega$):

  $$\forall x \in A, \forall y \in B : \int\limits_A \int\limits_B \pi\left(x\right) P_{MH}\left(x, y\right) dxdy = \int\limits_B \int\limits_A \pi\left(y\right) P_{MH}\left(y, x\right) dydx$$

  Let $x \in A$ and $y \in B$

  $$P_{MH}\left(x, B\right) = \int\limits_B a\left(x, y\right) q\left(x, y\right) dy + \left[\int\limits_\Omega 1 - a\left(x, z\right) q\left(x, z\right) dz\right] \delta_x\left(y\right)$$

  $$= \int\limits_B a\left(x, y\right) q\left(x, dy\right) + s(x)\delta_x\left(y\right)$$

  With $s(x) = \int_\Omega 1 - a\left(x, z\right) q\left(x, z\right) dz$
  For any Borel sets A and B in Ω, we have:

$$\int\limits_A \int\limits_B \pi\left(x\right) P_{MH}\left(x, y\right) dxdy = \int\limits_A \int\limits_B \pi\left(x\right) a(x, y) q(x, y) dydx + \int\limits_{A \bigcap B} 1_{y=x} \pi\left(x\right) s(x) dx$$

$$= \int\limits_A \int\limits_B \pi\left(x\right) a(x, y) q(x, y) dydx + \int\limits_{A \cap B} 1_{y=x} \pi\left(y\right) s(y) dy$$

And as:

$$\pi\left(x\right)a(x,y)q(x,y)=\alpha\beta^{n(x)}\gamma^{-m(U_r(x))}\min\left(1,\beta^{n(y)-n(x_i)}\gamma^{m(U_r(x_i))-m(U_r(y))}\right)q(x,y)$$

$$=\alpha\min\left(\beta^{n(x)}\gamma^{-m(U_r(x))},\beta^{n(y)}\gamma^{-m(U_r(y))}\right)q(x,y)$$

$$=\alpha\beta^{n(y)}\gamma^{-m(U_r(y))}\min\left(\beta^{n(x)-n(y)}\gamma^{m(U_r(y))-m(U_r(x))},1\right)q(x,y)$$

$$=\alpha\beta^{n(y)}\gamma^{-m(U_r(y))}\min\left(1,\beta^{n(x)-n(y)}\gamma^{m(U_r(y))-m(U_r(x))}\right)q(x,y)$$

And as $q(x,y)=q(y,x)$ so $\pi\left(y\right)a(y,x)q(y,x)=\pi\left(x\right)a(x,y)q(x,y)$
Using the Fubini's theorem, we obtain:

$$\int\limits_A\int\limits_B\pi\left(x\right)a(x,y)q(x,y)dydx=\int\limits_B\int\limits_A\pi\left(y\right)a(y,x)q(y,x)dxdy$$

Finally, we have:

$$\int\limits_A\int\limits_B\pi\left(x\right)P_{MH}\left(x,y\right)dxdy=\int\limits_B\int\limits_A\pi\left(y\right)a(y,x)q(y,x)dxdy+\int\limits_{A\bigcap B}1_{y=x}\pi\left(y\right)s(y)dy$$

$$=\int\limits_B\int\limits_A\pi\left(y\right)P_{MH}\left(y,x\right)dydx$$

Hence, P_{MH} is π-reversible.

- **π-stationarity**
  A distribution π is said to be stationary for the transition kernel P_{MH} if:

$$\pi\left(x\right)=\int\limits_\Omega\pi\left(y\right)P_{MH}\left(y,x\right)dy$$

$$\int\limits_\Omega\pi\left(y\right)P_{MH}\left(y,x\right)dy=\int\limits_\Omega\pi(y)\left[a\left(x,y\right)q\left(x,y\right)+\left[\int\limits_\Omega 1-a\left(x,z\right)q\left(x,z\right)dz\right]\delta_x\left(y\right)\right]dy$$

$$=\int\limits_\Omega\pi(y)a\left(x,y\right)q\left(x,y\right)dy+\int\limits_\Omega\int\limits_\Omega\delta_x\left(y\right)\pi(y)-\pi(y)a\left(x,z\right)q\left(x,z\right)\delta_x\left(y\right)dzdy$$

$$=\int\limits_\Omega\pi(y)a\left(x,y\right)q\left(x,y\right)dy+\int\limits_\Omega\int\limits_\Omega\delta_x\left(y\right)\pi(y)dzdy-\int\limits_\Omega\int\limits_\Omega\pi(y)a\left(x,z\right)q\left(x,z\right)\delta_x\left(y\right)dzdy$$

$$=\int\limits_\Omega\pi(y)a\left(x,y\right)q\left(x,y\right)dy+\int\limits_\Omega\delta_x\left(y\right)\pi(y)dy-\int\limits_\Omega\int\limits_\Omega 1_{x=y}\pi(y)a\left(x,z\right)q\left(x,z\right)dzdy$$

$$= \int_\Omega \pi(y)a\left(x,y\right)q\left(x,y\right)dy + \int_\Omega \delta_x\left(y\right)\pi(y)dy - \int_\Omega \pi(y)a\left(y,z\right)q\left(y,z\right)dz$$

$$= \int_\Omega \delta_x\left(y\right)\pi(y)dy = \pi\left(x\right)$$

So the chain admits π as a stationary distribution.

- **π-irreducibility**

  The transition P_{MH} is π-irreducible if:

  $$\forall A \in \mathcal{A}, \pi\left(A\right) > 0 \Rightarrow \exists\, t, P^t_{MH}\left(x,A\right) > 0$$

  For any Borel set A in $\mathcal{A}$ and for $t = 1$, we obtain:

  $$\int_\Omega 1_{B(x,A)}P_{MH}\left(x,A\right)dx = \int_\Omega 1_{B(x,A)}a\left(x,A\right)q\left(x,A\right)dx$$

  $$+ \int_\Omega 1_{B(x,A)}\left[1 - \int_\Omega a\left(x,z\right)q\left(x,z\right)dz\right]\delta_x\left(A\right)dx$$

  $$= \int_\Omega 1_{B(x,A)}a\left(x,A\right)q\left(x,A\right)dx + \int_\Omega 1_{B(x,x)}\left[1 - \int_\Omega a\left(x,z\right)q\left(x,z\right)dz\right]dx$$

  $$= \int_\Omega 1_{B(x,A)}a\left(x,A\right)q\left(x,A\right)dx + 1 - \int_\Omega\int_\Omega a\left(x,z\right)q\left(x,z\right)dzdx$$

  As $a(x,A) = \min\left(1, \gamma^{m(U_r(x))-m(U_r(A))}\right)$ and $a(x,z) = \min\left(1, \gamma^{m(U_r(x))-m(U_r(z))}\right)$, so there are four possible cases:

  - If $a\left(x,A\right) = 1$ and $a(x,z) = \gamma^{m(U_r(x))-m(U_r(z))}$ so:

    $$\int_\Omega 1_{B(x,A)}P_{MH}\left(x,A\right)dx = \int_\Omega 1_{B(x,A)}q\left(x,A\right)dx + 1 -$$

    $$\int_\Omega\int_\Omega \gamma^{m(U_r(x))-m(U_r(z))}q\left(x,z\right)dzdx$$

    $$= \int_\Omega 1_{B(x,A)}q\left(x,A\right)dx + 1 - \gamma^{m(U_r(x))-m(U_r(z))} > 0$$

  - if $a(x,A) = \gamma^{m(U_r(x))-m(U_r(A))}$and $a\left(x,z\right) = 1$ so:

$$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}(x,A)\,dx = \int\limits_{\Omega} 1_{B(x,A)} \gamma^{m(U_r(x))-m(U_r(A))} q(x,A)\,dx + 1$$

$$- \int\limits_{\Omega}\int\limits_{\Omega} q(x,z)\,dzdx$$

$$= \gamma^{m(U_r(x))-m(U_r(A))} \int\limits_{\Omega} 1_{B(x,A)} q(x,A)\,dx > 0$$

– If $a(x,A) = \gamma^{m(U_r(x))-m(U_r(A))}$ and $a(x,z) = \gamma^{m(U_r(x))-m(U_r(z))}$ so:

$$\int\limits_{\Omega} 1_{B(x,A)} P_{MH}(x,A)\,dx = \int\limits_{\Omega} 1_{B(x,A)} \gamma^{m(U_r(x))-m(U_r(A))} q(x,A)\,dx + 1$$

$$- \int\limits_{\Omega}\int\limits_{\Omega} \gamma^{m(U_r(x))-m(U_r(z))} q(x,z)\,dzdx$$

$$= \gamma^{m(U_r(x))-m(U_r(A))} \int\limits_{\Omega} 1_{B(x,A)} q(x,A)\,dx + 1 -$$

$$\gamma^{m(U_r(x))-m(U_r(z))} \int\limits_{\Omega}\int\limits_{\Omega} q(x,z)\,dzdx$$

$$= \gamma^{m(U_r(x))-m(U_r(A))} \int\limits_{\Omega} 1_{B(x,A)} q(x,A)\,dx + 1 - \ \gamma^{m(U_r(x))-m(U_r(z))} > 0$$

So $\int\limits_{\Omega} 1_{B(x,A)} {P_{MH}}^{t}(x,A)\,dx > 0 \ \forall\, t \geq 0$, hence, P_{MH} is π-irreducible.

By construction of $P = P_{MH}$, we have $\pi P = \pi$. If P is π-irreducible and possesses a π-invariant distribution, then P is positive recurrent and π is the unique invariant distribution of P [62] (see Proposition 1).

On the other hand, the chain created by the construction algorithm will also be aperiodic as long as at least one pair of configurations (x,y) exist, such that $a(x,y) < 1$, because then we will have $P(x,x) > 0$. We can quickly see that the chain is aperiodic as the event $X_{(N+1)} = X_{(N)}$ is possible practically at any time. Indeed, each state can be visited at two consecutive iterations, so $P^1(x,x) > 0$, making the period equal to 1. □

Theorem 5.1. *The Markov chain $(X_N)_{N\geq 0}$ obtained from the proposed construction algorithm* *is geometrically ergodic, and its kernel P realizes the simulation of the area-interaction point processes with density $\pi(x) = \alpha\beta^{n(x)}\gamma^{-m(U_r(x))}$. In other words, vP^n converges to π as n approaches infinity, where v is an initial distribution, and we have:*

$$\lim_{n\to\infty} \|vP^n - \pi\|_{TV} = 0$$

Here, $\|\cdot\|_{TV}$ denotes the total variation norm defined by:

$$\|\mu\|_{TV} = \sup_{|f|<1} |\mu(f)|$$

Proof. Let v be an initial distribution. For any integer n and $\forall\, x \in N^{lf}$, we have:

$$\|vP^n - \pi\|_{TV} = \|vP^n - \pi P^n\|_{TV} \leq \|v - \pi\|.\|P^n\| \leq 2c(P^n)$$

And we know that $c(P^n) \leq [c(P)]^n$ [64] (see Lemma 4.2.2 page 71), so $\|vP^n - \pi\|_{TV} \leq 2[c(P)]^n$. And $c(P)$ is the Dobrushin contraction coefficient [63] defined by the following:

$$c(P) = \frac{1}{2}\sup_{x,y} \|P(x,.) - P(y,.)\|_{TV}$$

According to Proposition 5.1, the kernel P is primitive, so $0 \leq c(P) < 1$ [64] (see Lemma 3.2). Therefore, as n tends to infinity, $\|vP^n - \pi\|_{TV}$ tends to zero. Hence, the chain is geometrically ergodic and converges to the distribution $\pi(x) = \alpha\beta^{n(x)}\gamma^{-m(U_r(x))}$. □

5.4 Convergence Speed of the Proposed Algorithm

The objective of this section is to study an approach to estimate the speed at which the Markov kernel $P = P_{MH}$, which is π-reversible, converges to the distribution π.

Definition 5.2. *A Markov kernel K on Ω is a mapping from Ω to Borel measures on Ω such that:*

- *For every $x \in \Omega$, $K(x, dy)$ is a probability measure.*
- *For any $\mathcal{A}$, $K(x, A) = \int\limits_A K(x, dy)$ is a measurable function.*
- *For $f \in L^\infty(\Omega)$, the Markov operator $K(f)$ is defined as follows:*

$$K(f)(x) = \int\limits_\Omega f(y)K(x, dy)$$

(Here, we denote $L^{\infty}(\Omega)$ as the space of bounded functions on Ω, equipped with the norm $\|f\|_{\infty} = \sup\limits_{x\in\Omega} |f(x)|$).

Definition 5.3. *Let π be a probability measure and a Markov operator, we say that is π-reversible, if and only if it is self-adjoint on, which means that:*

$$\langle f|g\rangle_{\pi} = \int K(f)\, g d\pi = \int f K(g)\, d\pi = \langle g|f\rangle_{\pi}, \forall f, g, \pi \in L^{\infty}(\Omega)$$

Theorem 5.2. *The Markov chain $(X_n)_{n\in\mathbb{N}}$ associated with P in the proposed algorithm satisfies the following properties for all n: P is irreducible and admits a reversible measure π. Therefore, the eigenvalues $\lambda_1 > \lambda_2 \cdots > \lambda_N$ of P satisfy $\lambda_1 = 1$. Additionally, for any initial distribution π_0, we have:*

$$\|\pi_n - \pi\|_{TV} \leq Ce^{-\rho n} \text{ with } C = \frac{1}{2}\sqrt{\max_{x\in\Omega}\left(\frac{1}{\pi(x)} - 1\right)}$$

where $\rho = \min(1 - |\lambda_2|, 1 - |\lambda_N|)$ is the spectral gap of P, and $\pi_n = \pi_0 P^n$ represents the marginal distribution of $(X_n)_{n\in\mathbb{N}}$, and P is exponentially ergodic and converges exponentially fast towards the target distribution π.

Proof. According to Proposition 5.1, P is π-irreducible, π-reversible, and primitive. By the Perron–Frobenius theorem [85], P has a strictly positive real eigenvalue λ_1, and $\lambda_1 > \lambda_2 \cdots > \lambda_N$. We need to show that these eigenvalues are contained in the interval $[-1, 1]$.

Let μ be an eigenvector of λ, then we have the following:

$$\left|\langle\mu|\mu P\rangle_{\frac{1}{\pi}}\right| = \left|\int\int \mu(x)\mu(y) P(x,y) \frac{dxdy}{\pi(x)}\right| = \left|\int\int \pi(y) \frac{\mu(x)}{\pi(x)} \frac{\mu(y)}{\pi(y)} P(x,y)\, dxdy\right|$$

$$\leq \frac{1}{2}\int\int \pi(y) \left(\frac{\mu(x)}{\pi(x)}\right)^2 P(x,y)\, dxdy + \frac{1}{2}\int\int \pi(y)\left(\frac{\mu(y)}{\pi(y)}\right)^2 P(x,y)\, dxdy$$

$$\leq \frac{1}{2}\int \pi(x)\left(\frac{\mu(x)}{\pi(x)}\right)^2 dx + \frac{1}{2}\int \pi(y)\left(\frac{\mu(y)}{\pi(y)}\right)^2 dy \leq \int (\mu(x))^2 \frac{dx}{\pi(x)} = \langle\mu|\mu\rangle_{\frac{1}{\pi}}$$

And as μ is an eigenvector of λ, we have $\left(\|\mu\|_{\frac{1}{\pi}}\right)^2 \geq |\lambda| \left(\|\mu\|_{\frac{1}{\pi}}\right)^2$, which implies $1 \geq |\lambda|$. Therefore, we have:

$$1 \geq \lambda_1 > \lambda_2 \cdots > \lambda_N \geq -1$$

If $\lambda = 1$, then:

$$\frac{\mu(x)}{\pi(x)}\frac{\mu(y)}{\pi(y)} = \frac{1}{2}\left(\frac{\mu(x)}{\pi(x)}\right)^2 + \frac{1}{2}\left(\frac{\mu(y)}{\pi(y)}\right)^2 \Rightarrow \frac{\mu(x)}{\pi(x)} = \frac{\mu(y)}{\pi(y)}$$

That is, $\mu = c\pi$, with c being a constant.

Reciprocally, from the definition of an invariant measure, π indeed satisfies $\pi = \pi P$. Thus, we have established that the eigenvalue 1 is a simple eigenvalue, which we denote as $\lambda_1 = 1$.

Now, let us show that -1 is not an eigenvalue of P. We assume by contradiction that there exists an eigenvector f for the eigenvalue -1. Let x be a state in the state space Ω, such that $f(x) > 0$ and $f(x) = \max_{y\in\Omega} |f(y)|$ (we can take $-f$ instead of f if necessary). As P is aperiodic according to Proposition 5.1, for sufficiently large n, $P^n(x,x) > 0$ (see Lemma 3.1). We have:

$$\begin{aligned} -f(x) = P^n(f)(x) &= \int P^n(x,y)\, f(y)dy \\ &> -\int P^n(x,y)\, |f(y)|\, dy \\ &> -\int P^n(x,y)\, f(x)dy = -f(x) \end{aligned}$$

This implies that $-f(x) > -f(x)$. It is a contradiction.

Finally, let us show that $\|\pi_n - \pi\|_{TV} \leq Ce^{-\rho n}$. Using the properties of the total variation norm [86] , we have the following:

$$2\|\pi_n - \pi\|_{TV} = \|\pi_n - \pi\|_1 \leq \|\pi_n - \pi\|_2$$

$$= \left(\int \left(\frac{\pi_n(x)}{\pi(x)} - 1\right)^2 \pi(x)\, dx\right)^{\frac{1}{2}} \leq \left(\int (\pi_n(x) - \pi_n(x))^2 \frac{dx}{\pi(x)}\right)^{\frac{1}{2}} = \|\pi_n - \pi\|_{\mathcal{L}^2(\Omega,\frac{1}{\pi})}$$

As P is a symmetric operator for its right action on $\mathcal{L}^2\left(\Omega, \frac{1}{\pi}\right)$, there exists an orthonormal basis $(\mu_1, \mu_2, \cdots, \mu_N)$ of this space corresponding to the eigenvalues $(1, \lambda_1, \cdots \lambda_N)$. The orthonormal eigenvector μ_1 for the eigenvalue 1 is simply the invariant measure π. Let us write the decomposition of the initial distribution π_0 in the eigenbasis of P:

$$\pi_0 = c\pi + \sum_{i=2}^{N} c_i \mu_i$$

For certain real coefficients c and c_i, $i \in [\![2, N]\!]$, with $c_i = \langle \pi_0 | \mu_i \rangle_{\frac{1}{\pi}}$, applying P^n to the formula above yields:

$$\pi_n = c\pi + \sum_{i=2}^{N} c_i (\lambda_i)^n \mu_i$$

As $|\lambda_i| < 1$ for $i \in [\![2, N]\!]$, then $\lim\limits_{n\to\infty} \pi_n = c\pi + \lim\limits_{n\to\infty} \sum\limits_{i=2}^{N} c_i(\lambda_i)^n \mu_i = c\pi$, which implies that $c = 1$. Therefore:

$$\pi_n - \pi = \sum_{i=2}^{N} c_i (\lambda_i)^n \mu_i$$

and

$$\left(\|\pi_n - \pi\|_{\frac{1}{\pi}}\right)^2 = \sum_{i=2}^{N} c_i^2 (\lambda_i)^{2n} \leq \sum_{i=2}^{N} c_i^2 (1-\rho)^{2n} \leq \left(\|\pi_0 - \pi\|_{\frac{1}{\pi}}\right)^2 e^{-2n\rho}$$

With $\rho = \min\left(1 - |\lambda_2|, 1 - |\lambda_N|\right)$ To conclude, we need to bound the value $\left(\|\pi_0 - \pi\|_{\frac{1}{\pi}}\right)^2$, which we denote as $F(\pi_0)$.

$$F(\pi_0) = \left(\|\pi_0 - \pi\|_{\frac{1}{\pi}}\right)^2 = \int (\pi_0(x) - \pi(x))^2 \frac{dx}{\pi(x)}$$

The function F is quadratic and, therefore, convex over a convex part of $\mathbb{R}^N$. Hence, it reaches its maximum at an extreme point of the convex set formed by probability measures. The extreme points of this convex set are precisely the Dirac measures δ_{x_0} (convex combination of Dirac measures). Thus,

$$\max_{\pi_0 \text{ probability on } \Omega} \left(\|\pi_0 - \pi\|_{\frac{1}{\pi}}\right)^2 = \max_{x_0 \in \Omega} \left(\|\delta_{x_0} - \pi\|_{\frac{1}{\pi}}\right)^2$$

For fixed x_0, we have the following:

$$\left(\|\delta_{x_0} - \pi\|_{\frac{1}{\pi}}\right)^2 = \frac{(1 - \pi(x_0))^2}{\pi(x_0)} + \int\limits_{\Omega} \pi(x)\, \delta_{x \neq x_0} dx$$

$$= \frac{(1 - \pi(x_0))^2}{\pi(x_0)} + (1 - \pi(x_0)) = \frac{1 + (\pi(x_0))^2 - 2\pi(x_0) + \pi(x_0) - (\pi(x_0))^2}{\pi(x_0)}$$

$$= \frac{1 - \pi(x_0)}{\pi(x_0)} = \frac{1}{\pi(x_0)} - 1$$

We conclude that $\|\pi_n - \pi\|_{TV} \leq Ce^{-\rho n}$ with $C = \frac{1}{2}\sqrt{\max_{x\in\Omega}\left(\frac{1}{\pi(x)} - 1\right)}$. □

5.5 Numerical Results and Quality of the Proposed Designs

To evaluate the quality of the proposed numerical experiment design, the same procedure as in section 4.4 is used.

Table 1 presents a comparison based on the discrepancy criterion between the designs proposed in this work (denoted AID: Area Interaction Design) and low discrepancy sequences (Halton sequence, Sobol sequence, and Faure sequence). It is interesting to observe that the proposed designs exhibit low discrepancy, comparable to that of low discrepancy sequences.

Tableau 5.1: Discrepancy value for the proposed designs (AID), Halton sequences, Sobol sequences, and Faure sequences for 2 and 3 factors.

Number of Factors	Number of Points	AID	Halton Sequence	Sobol Sequence	Faure Sequence
2	50	0.0005215	0.001076	0.000496	0.001076
3	100	0.0009192	0.000178	0.000112	0.000127

The constructed designs are also compared with commonly used designs in numerical experiments, excluding low discrepancy sequences. To give meaning to the results, the criteria were calculated for 100 designs. The designs compared in this section are as follows:

- Random Designs (RD)
- Latin Hypercube Designs (LHS)
- Maximin LHS Designs (mLHS)
- Strauss Designs (SD)
- Maximum Entropy Designs (Dmax)
- Marked Strauss Designs

- Connected Components Interaction Model Designs (CCD)
- Proposed Designs (TMD)

The figures below represent the results of the different criteria in the form of box plots for two and three dimensions.

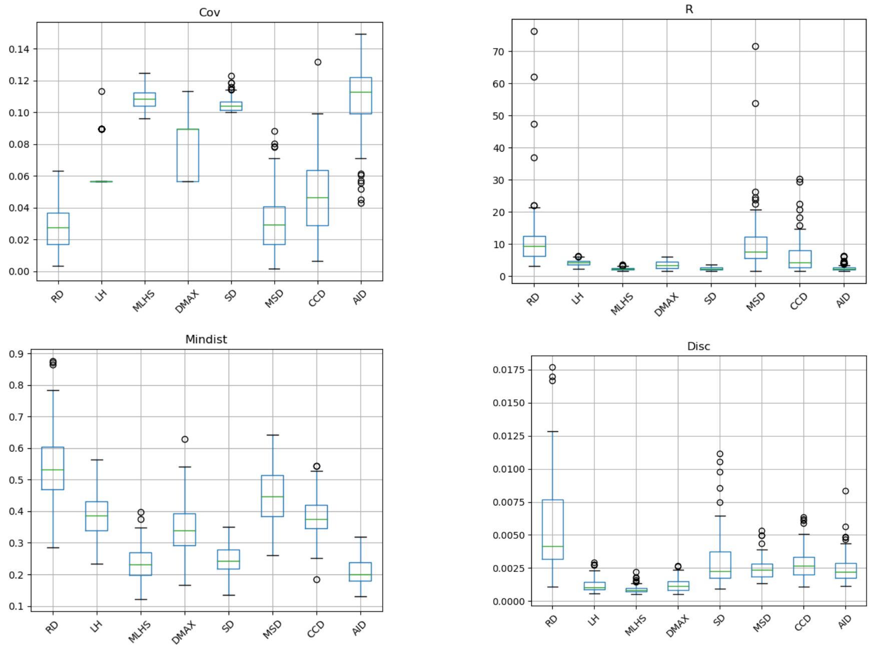

Figure 5.6: Box plots of the quality criteria calculated for 100 designs with 30 points in 5 dimensions.

According to the results presented in the box plots above, in two dimensions, the proposed designs show significant improvements both in terms of the ratio and the minimum distance criterion. This indicates that the points in the proposed designs are well separated from each other, ensuring good coverage of the experimental space, although there is no optimal overlap according to the distance criterion. Examining the discrepancy criterion, it can be confirmed that the proposed designs generate points that are more uniformly distributed within the unit cube compared to designs in the same category, such as Strauss designs, marked Strauss designs, and Connected Components Interaction designs. The proposed designs are also better than conventional Latin Hypercube Designs (LHD) and Maximin Latin Hypercube Designs in terms of distance criteria. In two dimensions, they appear to be a good compromise between a set of well-spread points in space and its projection on the margins.

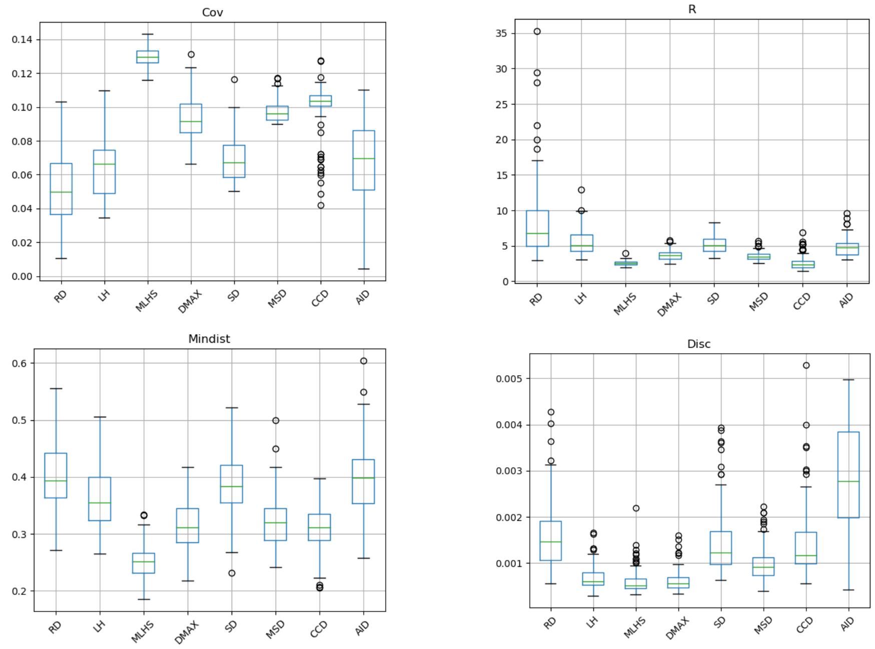

Figure 5.7: Box plots of quality criteria calculated for 80 designs with 50 points in 7 dimensions.

In three dimensions, the proposed designs and the Latin Hypercube Designs are better than the other designs in terms of distance criteria. In three dimensions, they visibly achieve a point distribution closer to a regular grid, ensuring good space coverage. It is surprising that the proposed designs do not perform as well in terms of discrepancy in three dimensions, especially compared to their performance in two dimensions. This may be explained by an inappropriate choice of interaction radii, which is compensated by potential power, avoiding points close to each other according to the usual distance. Thus, in three dimensions, the proposed designs generally have the worst discrepancy, but they stand out as the best in terms of distance criteria.

5.6 Conclusion

The use of point processes with area interaction, the MCMC method, and the Metropolis-Hastings algorithm with Voronoi tessellations allows for the construction of new numerical designs specific to this process. This approach offers great flexibility, as one can easily manipulate the distribution of the process via its representation to impose properties, such as optimal space filling and uniform distribution of points. Moreover, the experimental design method, combined with the use of point processes with area interaction and the MCMC methodology, provides an interesting alternative to the classical statistical approach of working with independent realizations from the same distribution. The designs constructed in this work have been compared with those commonly used in numerical experiments, and the results obtained are very satisfactory.

Finally, several perspectives can be envisaged, for example, it would be relevant to identify the issues related to points inside the infinite regions of Voronoi tessellations in a closed hypercube [87]. It would also be useful to find efficient numerical methods for calculating hypervolume for the union of fixed-radius hyperspheres.

General Conclusion

This thesis has thoroughly explored the application of stochastic processes to Computer Experiment Design plans, with a particular focus on the use of marked point processes and area-interaction point processes. In response to the challenges posed by the increasing complexity of modern simulators, our research has proposed innovative approaches to enhance the efficiency and accuracy of these experimental designs. By combining mathematical rigor with simulation innovations, we have opened new perspectives in the optimization of numerical simulations.

We demonstrated that integrating marked point processes with two marks, as well as area-interaction point processes, offers significant advantages for optimizing the distribution of points in the unit hypercube. These methods enable a better understanding of the complex interactions between experimental points, ensuring a more homogeneous and comprehensive coverage of the experimental domain. The use of the Metropolis-Hastings algorithm within the framework of Monte Carlo simulations via Markov chains (MCMC) has been pivotal in generating optimal configurations of experimental points. This allowed us to maximize the efficiency of numerical experimental designs while minimizing model-related errors, thus ensuring reliable and robust results that meet the requirements of contemporary simulations. The results obtained indicate that the new digital experimental designs proposed in this thesis meet the precision and reliability demands of complex simulators. In particular, designs based on area-interaction point processes demonstrated an enhanced ability to detect and manage irregularities, thereby providing more comprehensive and relevant insights for analyzing the modeled physical phenomena.

This research represents a significant extension of previous work on numerical experimental designs, particularly through the innovative application of two-marked point processes and area-interaction point processes. The contributions of this thesis provide valuable tools and methods for simulating complex processes, enabling a deeper understanding of the underlying dynamics of these systems. These methodological advances pave the way for new possibilities in various fields, including natural sciences, engineering, and economic and social studies, where precise and efficient simulations are crucial for decision-making and innovation.

Future work could focus on exploring other types of stochastic processes, including the development of a new type of point process [88] and the study of adding a third mark to create a three-marked experimental design [89]. This approach would offer a new dimension of marking, further enriching digital experimental designs and enabling better coverage of the experimental domain. Additionally, this new dimension could refine the management of interactions between experimental points, thereby increasing the accuracy of the obtained results. Another promising avenue would be to apply these digital experimental designs to hyperparameter optimization in machine learning models, a field where simulation efficiency and optimal resource management are critical. Moreover, integrating these methods into more powerful computing environments and validating their performance in practical scenarios could open new perspectives for the evolution of numerical experimental designs, providing even more robust solutions tailored to current challenges.

In conclusion, this thesis has not only deepened our understanding of stochastic processes in the context of digital experimental designs but also laid the groundwork for new perspectives in the optimization and management of complex simulators. The approaches developed here establish a solid foundation for future research and potential applications in various scientific and industrial domains. These works thus pave the way for innovations that could transform current practices in numerical simulation, with significant impacts across a wide range of sectors.

Appendix A

Python Code for the Results Presented in Chapter 4

Optimality Criteria

Distance Criterion

```python
def mindist(x):
    n = x.shape[0]  # Nombre de points dans x
    w = x.shape[1]  # Dimension de x
    M = np.zeros((n, n))
    for i in range(n - 1):
        for k in range(i + 1, n):
            s = 0
            for j in range(w):
                s += (x[i, j] - x[k, j]) ** 2
            M[i, k] = np.sqrt(s)
            M[k, i] = np.sqrt(s)

    for i in range(n):
        M[i, i] = np.inf
    y = np.min(np.min(M))
    return y
```

Discrepancy Criterion

```python
def dsc(x):
    n = x.shape[0]  # Nombre de points dans x
    w = x.shape[1]  # Dimension de x
    s1 = 0
    for i in range(n):
        p1 = 1
        for j in range(w):
            p1 *= (1 - x[i, j]) * (1 + x[i, j])
        s1 += p1
    s2 = 0
    for i in range(n):
        for j in range(n):
            p2 = 1
            for k in range(w):
                m = max(x[i, k], x[j, k])
                p2 *= (1 - m)
            s2 += p2
    y = (3 ** (-w)) + (1 / (n ** 2)) * s2 - (1 / (n * 2 ** (w - 1))) *
        s1
    return y
```

Coverage Criterion

```
def mdist(x):
    n = x.shape[0]  # Nombre de points dans x
    w = x.shape[1]  # Dimension de x
    M = np.zeros((n, n))
    for i in range(n - 1):
        for k in range(i + 1, n):
            s = 0
            for j in range(w):
                s += (x[i, j] - x[k, j]) ** 2
            M[i, k] = np.sqrt(s)
            M[k, i] = np.sqrt(s)
    for i in range(n):
        M[i, i] = np.inf
    t = np.zeros(n)
    for i in range(n):
        t[i] = np.min(M[i, :])
    q = np.sum(t)
    q1 = q / n
    lamda = 0
    for i in range(n):
        lamda += (t[i] - q1) ** 2
    y = (1 / q1) * ((1 / n) * lamda) ** 0.5
    return y
```

R Criterion

```
    def rap(X):
    n = X.shape[0]
    w = X.shape[1]
    M = np.zeros((n, n))
    for i in range(n - 1):
        for k in range(i + 1, n):
            s = 0
            for j in range(w):
                s += (X[i, j] - X[k, j]) ** 2
            M[i, k] = np.sqrt(s)
            M[k, i] = np.sqrt(s)
    np.fill_diagonal(M, np.inf)
    t = np.min(M, axis=1)
    y = np.ptp(t) / np.min(t)
    return y
```

Metropolis-Hasting Algorithm and exporting the Distributions

```
import numpy as np
from scipy.integrate import quad
import pandas as pd

w=10       # dimention
r=0.05               # rayon d'interaction
# Definition des parametres du modele
n = 128         # nombre de points
eps = 0.07
R = 0.1              # rayon de la gaussienne de proposition
g11 =0.01
g12 = 0.01             # coefficient d'interaction pour les paires de
   marque (1,2)
```

```
g22 = 0.05              # coefficient d'interaction pour les paires de
    marque (2,2)
b1 = 0.9                #itensite du processus pour la marque 1
b2 = 1.5                #itensite du processus pour la marque 1
NMC = 1000
#marques1
def m1(X):
    n = X.shape[0]
    m = np.zeros(n)
    XTX_inv = np.linalg.inv(np.matmul(X.T, X))
    for i in range(n):
        m[i] = np.matmul(np.matmul(X[i, :], XTX_inv), X[i, :].T)
    return m
#marques2
def distance(x1, x2):
    return np.linalg.norm(x1-x2)
def density(x):
    return np.exp(-(x)**2/2) / (np.sqrt(2*np.pi))
def m2(X):
    n = X.shape[0]
    mus = np.zeros(n)
    for i in range(n):
        sum_distances = 0
        for j in range(n):
            if i != j:
                dist = distance(X[i], X[j])
                integral, _ = quad(lambda t: density(t), 0, dist)
                sum_distances += integral
        mus[i] = sum_distances / (n-1)
    return mus
def m11(X, m1,r, eps):
    n = X.shape[0]
    count = 0
    for i in range(n):
        for j in range(i+1, n):
            if np.linalg.norm(X[i] - X[j]) <= r and m1[i] <= eps and
               m1[j] <= eps:
                count += 1
    return count
def m22(X, m1, eps,r):
    n = X.shape[0]
    count = 0
    for i in range(n):
        for j in range(i+1, n):
            if np.linalg.norm(X[i] - X[j]) <= r and m1[i] >= R and m1[
               j]  >= R:
                count += 1
    return count
def m12(X, m1, m2, eps,r, R):
    n = X.shape[0]
    count = 0
    for i in range(n):
        for j in range(i+1, n):
            if  m1[i] <= eps and m1[j] <= eps and m2[i] >= R and m2[j
               ]  >= R:
                dist = np.linalg.norm(X[i] - X[j])
                if dist <= r:
                    count += 1
    return count
# Generation d'une configuration initiale de points aleatoires dans un
    carre unite
# Create empty DataFrames to store values
df_a = pd.DataFrame(columns=['Valeur'])
df_b = pd.DataFrame(columns=['Valeur'])
df_c = pd.DataFrame(columns=['Valeur'])
df_d = pd.DataFrame(columns=['Valeur'])
# Execute the code 20 times
# Simulation de NMC etapes
for _ in range(1):
   X = np.random.rand(n, w)
   for N in range(NMC):
       print("Itera", _ + 1,N+1)
# choix d'un point au hasard
       k = np.random.randint(n)
    # creation d'une nouvelle configuration Y
       y = np.random.rand(1, w)
       Y = np.copy(X)
       Y[k] = y
    # Calcul des valeurs de m1, m2, m11, m22 et m12 pour la
       configuration actuelle
       m1_vals = m1(X)
       num_points_m1 = len(np.where(m1_vals <= eps)[0])
       m2_vals = m2(X)
       num_points_m2 = len(np.where(m2_vals >= eps)[0])
       m11_vals = m11(X, m1_vals, r,eps)
       m22_vals = m22(X, m2_vals, r,R)
       m12_vals = m12(X, m1_vals, m2_vals,eps,r, R)
    # Calcul des valeurs de m1, m2, m11, m22 et m12 pour la nouvelle
       configuration
       m1_vals_new = m1(Y)
       num_points_m1_new = len(np.where(m1_vals_new <= eps)[0])
       m2_vals_new = m2(Y)
       num_points_m2_new = len(np.where(m2_vals_new >= eps)[0])
       m11_vals_new = m11(Y, m1_vals, R,eps)
       m22_vals_new = m22(Y, m2_vals_new,r, R)
       m12_vals_new = m12(Y, m1_vals_new, m2_vals_new,eps,r, R)
     # calcul de la probabilite d'acceptation
       by = b1**num_points_m1_new *b2**num_points_m2_new*g11**
          m11_vals_new*g12**m12_vals_new*g22**m22_vals_new
       bx = b1**num_points_m1*b2**num_points_m2*g11**m11_vals*g12**
          m12_vals*g22**m22_vals
       a = min(1, by / bx)
    #print(a)
    #mise a jour de la configuration
       if a ==1:
          X[k] = y
     # Calcul de la valeur de a et b
   a = mdist(X)
   b = mindist(X)
   c = dsc(X)
   d=rap(X)
   print(c)
      # Store the results in DataFrames
 #  df_a = pd.concat([df_a, pd.DataFrame({'Valeur': [a]})],
    ignore_index=True)
 #  df_b = pd.concat([df_b, pd.DataFrame({'Valeur': [b]})],
    ignore_index=True)
 #  df_c = pd.concat([df_c, pd.DataFrame({'Valeur': [c]})],
    ignore_index=True)
 #  df_d = pd.concat([df_d, pd.DataFrame({'Valeur': [d]})],
    ignore_index=True)
  # Save the DataFrames to a single Excel file
#with pd.ExcelWriter('output50_7d.xlsx') as writer:
 #    df_a.to_excel(writer, sheet_name='mdist', index=False)
  #    df_b.to_excel(writer, sheet_name='mindist', index=False)
  #    df_c.to_excel(writer, sheet_name='dsc', index=False)
  #    df_d.to_excel(writer, sheet_name='R', index=False)
```

Appendix B

Python Code for the Results Presented in Chapter 5

Results in 2 Dimensions

Libraries

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial import Voronoi, voronoi_plot_2d
import random
from matplotlib.patches import Polygon
import math
from matplotlib.path import Path
from matplotlib.patches import Circle
from shapely.geometry import Point
from shapely.ops import cascaded_union, unary_union
from matplotlib.path import Path
```

Voronoi Functions

```python
def voronoi_finite_polygons_2d(vor, radius=None):
    # Function implementation goes here
    if vor.points.shape[1] != 2:
        raise ValueError("Requires 2D input")

    new_regions = []
    new_vertices = vor.vertices.tolist()
    center = vor.points.mean(axis=0)
    if radius is None:
        radius = vor.points.ptp().max() * 2

    # Construct a map containing all ridges for a given point
    all_ridges = {}
    for (p1, p2), (v1, v2) in zip(vor.ridge_points, vor.ridge_vertices
       ):
        all_ridges.setdefault(p1, []).append((p2, v1, v2))
        all_ridges.setdefault(p2, []).append((p1, v1, v2))

    # Reconstruct infinite regions
    for p1, region in enumerate(vor.point_region):
        vertices = vor.regions[region]

        if all(v >= 0 for v in vertices):
            # finite region
            new_regions.append(vertices)
            continue

        # reconstruct a non-finite region
        ridges = all_ridges[p1]
        new_region = [v for v in vertices if v >= 0]

        for p2, v1, v2 in ridges:
            if v2 < 0:
                v1, v2 = v2, v1
            if v1 >= 0:
                # finite ridge: already in the region
                continue

            # Compute the missing endpoint of an infinite ridge
            t = vor.points[p2] - vor.points[p1]  # tangent
            t /= np.linalg.norm(t)
            n = np.array([-t[1], t[0]])  # normal

            midpoint = vor.points[[p1, p2]].mean(axis=0)
            direction = np.sign(np.dot(midpoint - center, n)) * n
            far_point = vor.vertices[v2] + direction * radius

            new_region.append(len(new_vertices))
            new_vertices.append(far_point.tolist())

        # sort region counterclockwise
        vs = np.asarray([new_vertices[v] for v in new_region])
        c = vs.mean(axis=0)
        angles = np.arctan2(vs[:, 1] - c[1], vs[:, 0] - c[0])
        new_region = np.array(new_region)[np.argsort(angles)]

        # finish
        new_regions.append(new_region.tolist())

    return new_regions, np.asarray(new_vertices)

def voronoi_function(X):
    # Create Voronoi tessellation and plot it
    vor = Voronoi(X)

    # Apply the voronoi_finite_polygons_2d function
    new_regions, new_vertices = voronoi_finite_polygons_2d(vor)
    # Iterate over each region and add a point
    new_points = []
    for region in new_regions:
        polygon = [new_vertices[i] for i in region]
        path = Path(polygon)
        while True:
            random_point = np.random.uniform(0, 1, 2)
            if path.contains_point(random_point):
                new_points.append(random_point.tolist())
                break
    # Convert the new points to a NumPy array
    new_points = np.array(new_points)
    # Add the new points to the original set of points
    X = np.vstack((X, new_points))

    # Randomly select and remove one of two given points with 50% probability
    while len(X) > n:
        indices = np.arange(len(X))
        chosen_points = set()
        i = np.random.choice(indices)
        if i in chosen_points:
            continue
        chosen_points.add(i)
        if i >= len(X):
            continue
        distances = np.linalg.norm(X - X[i], axis=1)
        distances[i] = np.inf
        min_distance_index = np.argmin(distances)
        indices = np.delete(indices, min_distance_index)
        X = np.delete(X, min_distance_index, axis=0)

    return X
```

Area Interaction Functions

```
def calculate_overlap(points , radius):
    circles = [Point(point).buffer(radius) for point in points]
    union = unary_union(circles)
    total_area = union.area
    return total_area
```

Metropolis-Hasting Algorithm and plot of the Initial and Final configurations

```
n = 25          #Number of points
N = 10000       #Number of iterations of MH
r = 0.095       #Radius
gamma =1.5
X = np.random.uniform(size=(n, 2))
X_intial = X.copy()
X_final = X.copy()
 # Plot the initial configuration
fig1, ax1 = plt.subplots()
ax1.scatter(X[:, 0], X[:, 1], color=’blue’, label=’Initial
   Configuration’)
    # Draw circles on the initial points
for point in X:
    circle = Circle((point[0], point[1]), r, color=’red’, fill=False)
    ax1.add_patch(circle)
ax1.set_aspect(’equal’)
ax1.set_title(’Initial Configuration’)
# Set x and y limits
ax1.set_xlim([-0.01, 1.01])
ax1.set_ylim([-0.01, 1.01])
# Calculate and print the initial area
area = calculate_overlap(X, r)
print("The initial area is:", area)
for iteration in range(N):
    X_intial = X
    area = calculate_overlap(X, r)
     #print("Iteration:", iteration + 1)
    # Perform the voronoi_function
    X = voronoi_function(X)
    # Calculate and print the final area
    area1 = calculate_overlap(X, r)
    if gamma >1:
       # Acceptance probability
        a = min(1, gamma ** (area1 - area))
      # print("The probability of acceptance is:", a)
    else:
        a = min(1, gamma ** (area - area1))
    if a == 1:
        #print("We accept the final Configuration")
        X_final = X
       # print("the accepeted area is:",area1)
    else:
        #print("Playing Daft Punk: ONE MORE TIME")
        X_final = X_intial
       # print("the accepeted area is :",area)
    #print(’a1=’,area1,’a0=’,area)
    X = X_final
    area_final = calculate_overlap(X,r)
print("The final area is:", area_final)
# Plot the final configuration
fig2, ax2 = plt.subplots()
ax2.scatter(X[:, 0], X[:, 1], color=’blue’, label=’Final Configuration
   ’)
# Draw circles on the final points
for point in X:
    circle = Circle((point[0], point[1]), r, color=’red’, fill=False)
    ax2.add_patch(circle)
ax2.set_aspect(’equal’)
ax2.set_title(’Final Configuration’)
# Set x and y limits
ax2.set_xlim([-0.01, 1.01])
ax2.set_ylim([-0.01, 1.01])
plt.show()
```

B.0.0.1 Box plot of mdist

```
import pandas as pd
import matplotlib.pyplot as plt
# List of paths for the Excel files to import
file_paths = [r’C:\Users\StarTech\Desktop\mdist2dim_25p.xlsx’,
              r’C:\Users\StarTech\Desktop\mdist2dim_50p.xlsx’,
              r’C:\Users\StarTech\Desktop\mdist2dim_100p.xlsx’]
# Create a list to store the data from the Excel files
data = []
# Import the Excel data into the ’data’ list
for path in file_paths:
    data.append(pd.read_excel(path))
# Create a figure and an axis for the box plot
fig, ax = plt.subplots()
# Plot a box plot for each dataset
for i, dataset in enumerate(data):
    ax.boxplot(dataset.values, positions=[i+1])
# Set the x-axis labels
ax.set_xticks(range(1, len(file_paths)+1))
ax.set_xticklabels([’25 points’, ’50 points’, ’100 points’], rotation
   =45)
# Add a title to the graph
plt.title(’Mdist’)
# Display the graph
plt.show()
```

General functions of the 3D Algorithm

Libraries

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.spatial import Voronoi
import random
import pandas as pd
```

Voronoi Function

```
def voronoi_function(points):
    vor = Voronoi(points)
    infinite_regions = [region for region in vor.regions if -1 in
       region]
    bounded_regions = []
    for region in infinite_regions:
        region_points = vor.vertices[region]
        min_bound = np.min(region_points, axis=0)
        max_bound = np.max(region_points, axis=0)
        bounded_regions.append((min_bound, max_bound))
    new_points = []
    for bounds in bounded_regions:
        min_bound, max_bound = bounds
        random_point = np.random.uniform(low=min_bound, high=max_bound
           )
        new_points.append(random_point)
    finite_regions = [region for region in vor.regions if region and
       -1 not in region]
    for region in finite_regions:
        region_points = vor.vertices[region]
        min_bound = np.min(region_points, axis=0)
        max_bound = np.max(region_points, axis=0)
        random_point = np.random.uniform(low=min_bound, high=max_bound
           )
        new_points.append(random_point)
    all_points = np.vstack((points, new_points))
    X = np.clip(all_points, 0, 1)
    while len(X) > n:
        indices = np.arange(len(X))
        chosen_points = set()
        i = np.random.choice(indices)
        if i in chosen_points:
            continue
        chosen_points.add(i)
        if i >= len(X):
            continue
        distances = np.linalg.norm(X - X[i], axis=1)
        distances[i] = np.inf
        min_distance_index = np.argmin(distances)
        indices = np.delete(indices, min_distance_index)
        X = np.delete(X, min_distance_index, axis=0)
    return X
```

Calculating Volume Interaction

```
def calculate_union_volume(points, radius, num_samples=1000000):
    min_point = np.min(points, axis=0) - radius
    max_point = np.max(points, axis=0) + radius
    random_points = np.random.uniform(min_point, max_point, size=(
        num_samples, 3))
    distances = np.linalg.norm(random_points[:, np.newaxis] - points,
        axis=-1)
    total_volume = np.sum(np.any(distances <= radius, axis=-1))
    bounding_box_volume = np.prod(max_point - min_point)
    volume_fraction = total_volume / num_samples
    total_volume = volume_fraction * bounding_box_volume
    return total_volume
```

Metropolis-Hasting Algorithm and exporting the Distributions

```
n = 25
N = 1000
r = 0.1
gamma =2
# Create empty DataFrames to store values
df_a = pd.DataFrame(columns=['Value'])
df_b = pd.DataFrame(columns=['Value'])
df_c = pd.DataFrame(columns=['Value'])
# Execute the code 20 times
'''results_a = []
results_b = []
results_c= []'''
for _ in range(100):
    #Generating initial configuration in the unit cube
    X = np.random.uniform(size=(n, 3))
    X_initial = X.copy()
    X_final = X.copy()
    area_intial = calculate_union_volume(X, r)
    print("The initial area is:", area_intial)
    # Simulation of NMC steps
    for iteration in range(N):
        X_intial = X
        area = calculate_union_volume(X, r)
        print("Iteration:", iteration + 1)
        # Perform the voronoi_function
        X = voronoi_function(X)
    # Calculate and print the final area
        area1 = calculate_union_volume(X, r)
        if gamma >1:
```

```
        # Acceptance probability
            a = min(1, gamma ** (area1 - area))
        # print("The probability of acceptance is:", a)
        else:
            a = min(1, gamma ** (area - area1))
        if a == 1:
        #print("We accept the final Configuration")
            X_final = X
            area = area1
       # print("the accepeted area is:",area1)
        else:
        #print("Playing Daft Punk: ONE MORE TIME")
            X_final = X_intial
       # print("the accepeted area is :",area)
    #print('a1=',area1,'a0=',area)
        X = X_final
    area_final = calculate_union_volume(X,r)
    print("The final area is:", area_final)
   # Calculate the values of a,b and c
    a = mdist(X)
    b = mindist(X)
    c = dsc(X)
    df_a = pd.concat([df_a, pd.DataFrame({'Value': [a]})],
       ignore_index=True)
    df_b = pd.concat([df_b, pd.DataFrame({'Value': [b]})],
       ignore_index=True)
    df_c = pd.concat([df_c, pd.DataFrame({'Value': [c]})],
       ignore_index=True)
# Export the DataFrames to the same Excel file
    file_name_a = 'mdist3dim_25p.xlsx'
    file_name_b = 'mindist3dim_25p.xlsx'
    file_name_c = 'disc3dim_25p.xlsx'
    df_a.to_excel(file_name_a, index=False)
    df_b.to_excel(file_name_b, index=False)
    df_c.to_excel(file_name_c, index=False)
    print(f'Result of  "a" Exported to {file_name_a}')
    print(f'Result of  "b" Exported to {file_name_b}')
    print(f'Result of  "c" Exported to {file_name_c}')
    results_a.append(a)
    results_b.append(b)
    results_c.append(c)
#Creat a DataFrame for each result
df_a = pd.DataFrame({'Value': results_a})
df_b = pd.DataFrame({'Value': results_b})
```

```
df_c = pd.DataFrame({'Value': results_c})
# Export DataFrames to separated Excel files
file_name_a = 'mdist7dim_35p.xlsx'
file_name_b = 'mindist7dim_35p.xlsx'
file_name_c = 'disc7dim_35p .xlsx'
df_a.to_excel(file_name_a, index=False)
df_b.to_excel(file_name_b, index=False)
df_b.to_excel(file_name_c, index=False)
print(f'Result of  "a" Exported to {file_name_a}')
print(f'Result of  "b" Exported to {file_name_b}')
print(f'Result of  "b" Exported to {file_name_c}')
```

Appendix C

List of Symbols and Abbreviations

$\mathcal{A}$	: A sigma-algebra on $\mathcal{X}$.
A, A_i	: Borel sets.
$a(.,.)$	: Probability of acceptance of change.
α	: Normalization constant.
β	: Vector of unknown coefficients, the intensity of the process.
$\hat{\beta}$	: Vector of estimated coefficients.
B	: Borel set.
$B\left(x_i, 2^{-j}\right)$	: Open ball.
B_n	: Ball centered at the origin.
$B(n, p)$	: Binomial distribution with parameters n and p.
$\mathrm{C}ov(.,.)$	: Mathematical covariance.
Cov	: Coverage.
$C\left(.\right), c\left(.\right)$	: Contraction coefficient.
C_n^m	: Combination of n points chosen from m points.
$\delta\left(.,.\right)$	: Normal density distance.
$\delta_x\left(.\right)$	: Dirac measure at x.
σ^2	: Variance of residuals.
$d^2\left(\hat{Y}\right)$	: Variance function of prediction.

$d\left(\hat{Y}\right)$	: Prediction error function.
d	: Metric on the configuration space.
d'	: Metric on the mark space.
$\partial\left(.\right)$	: Neighborhood.
∂	: Partial derivatives.
$Disc$	: Discrepancy.
$\varepsilon(x)$	: Residual vector.
$\mathbb{E}(.)$	: Mathematical expectation.
E^{lf}	: Family of locally finite configurations.
$f(x)$	: The response law at x.
F	: Fisher value.
$F_{1-\alpha,p,n-p-1}$	: Tabulated value.
γ_{lk}	: Interaction coefficients.
$\phi(y)$	: Interaction potentials.
$\Phi_b(i)$	: Inverse radical function.
I_n	: Identity matrix.
$Im(X)$	: Vector subspace spanned by the columns of X.
$(ImX)^{\perp}$	: Orthogonal subspace.
$j_n\left(x_1, x_2, \cdots, x_n\right)$	: Family of discrete distributions.
k	: Mark index.
$\mathcal{K}$	: Mark space.
K	: Markov transition kernel.
$K(f)$	: Markov operator.
λ	: Intensity of a homogeneous Poisson process.
$\lambda(u,x)$	: Papangelou conditional intensity.
MS	: Mean sum of squares.

$Mindist$	: Distance criterion.
$\mu(.)$	: Mean of normal density distances.
m	: Lebesgue measure.
m_k	: Number of points with mark k.
m_{lk}	: Number of point pairs of type (l, k) or (k, l).
n	: Number of experiments or number of points in a configuration.
η_A	: Number of visits to A.
$\mathbb{N}^p$	: k-dimensional natural number space.
N^{lf}	: Family of all locally finite configurations.
N_s^{lf}	: Set of locally finite configurations with simple points.
$\mathcal{N}^{lf}$	: Sigma-algebra.
$N_X(A)$	: Number of points in A as a discrete random variable.
$n(x)$	: Number of points in the configuration x.
Ω	: Universe.
$\mathcal{P}(X, .)$	: Probability measure.
P_{MH}	: Metropolis-Hastings transition kernel.
$P(x, y)$	: Transition kernel of the algorithm.
π	: Law of a random process.
$\pi(f)$	The expected value of the function f under the distribution pi.
Q	: Proposal transition kernel.
R	: Coverage ratio.
$\mathbb{R}^k$	: k-dimensional Euclidean space.
r	: Radius.
$r(x, y)$	: Metropolis-Hastings ratio.
SST	: Total sum of squares.
SSR	: Sum of squares due to regression.

SSE	: Sum of squares due to error.
$s(x)$	: Number of related point pairs in the configuration x.
τ_A	: Hitting time of set A.
$U_r(x)$	: Union of balls centered at x_i with radius r.
v	: Borel measure.
$\mathbb{V}(.)$	: Mathematical variance.
x_i	: Variable or factor.
x	: Configuration or design matrix.
$\left({}^tXX\right)^{-1}$	: Dispersion matrix.
$\mathcal{X}$	: Set of countable configurations with n points.
$(X_n)_{n\in\mathbb{N}}$	: Sequence of discrete random variables.
y	: Response or quantity of interest.
$\hat{y}_{x_i}$	: Prediction error at point x_i.
z	: Origin variable.
z_0	: Mean of high and low levels.
$(\Omega, \mathcal{A}, \mathbb{P})$	: Probabilistic space.
$\sim$	: Binary neighborhood relation.

Bibliography

[1] Fisher R., The Design of Experiments, 2nd Edition, Olivier and Boyd, Edinburgh, 1935.

[2] Kiefer J., Optimum experimental designs, Journal of the Royal Statistical Society: Series B (Methodological) 21 (2) (1959) 272–304.

[3] Box G. E. and Behnken D. W., Some new three level designs for the study of quantitative variables, Technometrics 2 (4) (1960) 455–475.

[4] Daley D. J. and Vere-Jones D., An introduction to the theory of point processes: volume I: elementary theory and methods, Springer, 2003.

[5] Van Lieshout M., Markov point processes and their applications, World Scientific, 2000.

[6] Baddeley A. and Møller J., Nearest-neighbour markov point processes and random sets, International Statistical Review/Revue Internationale de Statistique (1989) 89–121.

[7] Baddeley A. J. and Van Lieshout M., Area-interaction point processes, Annals of the Institute of Statistical Mathematics 47 (1995) 601–619.

[8] Chib S. and Greenberg E., Understanding the metropolis-hastings algorithm, The american statistician 49 (4) (1995) 327–335.

[9] Hastings W. K., Monte carlo sampling methods using markov chains and their applications, Biometrika 57 (1) (1970) 97–109.

[10] Sibson R., The dirichlet tessellation as an aid in data analysis, Scandinavian Journal of Statistics (1980) 14–20.

[11] Schimmerling P., Sisson J. C. and Zaïdi A., Pratique des plans d'expériences, TEC & DOC, 1998.

[12] Tinsson W., Plans d'expérience: constructions et analyses statistiques, Vol. 67, Springer Science & Business Media, 2010.

[13] Taguchi G. and Wu Y., Introduction to off-line quality control, central japan quality control association, Available from American Supplier Institute 32100 (1980) np.

[14] Goupy J., plans d'expérience: Les mélanges, Dunod, 2000.

[15] Loeza-Serrano S. I., Optimal Statistical Design for Variance Components in Multistage Variability Models, The University of Manchester (United Kingdom), 2014.

[16] Gauchi J.-P., Plans d'expériences optimaux pour modèles non linéaires, Plans d'expériences-Applications à l'entreprise (1997) np.

[17] Goupy J., Modélisation par les plans d'expériences, Techniques de l'ingénieur. Mesures et contrôle (R275) (2000) R275–1.

[18] Goupy J., La méthode des plans d'expériences: optimisation du choix des essais & de l'interprétation des résultats, Vol. 29, Dunod Paris, 1988.

[19] Goupy J., Plans d'expériences pour surfaces de réponse, Dunod, 1999.

[20] Mathieu D., Nony J. and Phan-Tan-Luu R., New efficient methodology for research using optimal design (nemrodw) software, lprai, univ, Aix-Marseille III, France 22.

[21] Dodge Y. and Rousson V., Analyse de régression appliquée, Dunod, 1999.

[22] Box G. E. and Hunter J., The 2 k—p fractional factorial designs part ii., Technometrics 3 (4) (1961) 449–458.

[23] Plackett R. L. and Burman J. P., The design of optimum multifactorial experiments, Biometrika 33 (4) (1946) 305–325.

[24] Souvay P., Plans d'expériences: méthode Taguchi, Vol. 23, Ed. Techniques Ingénieur, 2002.

[25] Box G. E., Empirical model building and response surfaces, John Wiley & Sons google schola 3 (1987) 27–37.

[26] Morris M. D., A class of three-level experimental designs for response surface modeling, Technometrics 42 (2) (2000) 111–121.

[27] MATHIEU D. and Roger P.-T.-L., Planification d'expériences en formulation: criblage, Ed. Techniques Ingénieur, 2000.

[28] Myers R. H., Montgomery D. C. and Anderson-Cook C. M., Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[29] Mitchell T. J., An algorithm for the construction of "d-optimal" experimental designs, Technometrics 42 (1) (2000) 48–54.

[30] Pronzato L. and Müller W. G., Design of computer experiments: space filling and beyond, Statistics and Computing 22 (2012) 681–701.

[31] Loh W.-L., On latin hypercube sampling, The annals of statistics 24 (5) (1996) 2058–2080.

[32] Halton J. H., On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numerische Mathematik 2 (1960) 84–90.

[33] Sobol I. M., Uniformly distributed sequences with an additional uniform property, USSR Computational mathematics and mathematical physics 16 (5) (1976) 236–242.

[34] Faure H., Discrépance de suites associées à un système de numération (en dimension s), Acta arithmetica 41 (4) (1982) 337–351.

[35] Faure H., Discrépance quadratique de la suite de van der corput et de sa symétrique, Acta Arithmetica 55 (4) (1990) 333–350.

[36] Johnson M. E., Moore L. M. and Ylvisaker D., Minimax and maximin distance designs, Journal of statistical planning and inference 26 (2) (1990) 131–148.

[37] Gunzburger M. and Burkardt J., Uniformity measures for point sample in hypercubes, Rapp. tech. Florida State University (cf. p. 73).

[38] Warnock T. T., Computational investigations of low-discrepancy point sets ii, in: Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing, Springer New York, New York, NY, 1995, pp. 354–361.

[39] Chiu S. N., Stoyan D., Kendall W. S. and Mecke J., Stochastic geometry and its applications, John Wiley & Sons, 2013.

[40] Choiruddin A., Sélection de variables pour des processus ponctuels spatiaux, Ph.D. thesis, Université Grenoble Alpes (ComUE) (2017).

[41] Vere-Jones D. and Daley D., An introduction to the theory of point processes, Springer Ser. Statist., Springer, New York.

[42] Gaetan C. and Guyon X., Modélisation et statistique spatiales, Vol. 11, Springer, 2008.

[43] Barndorff-Nielsen O. E., Kendall W. S. and Van Lieshout M., Stochastic geometry: likelihood and computation, Chapman & Hall/CRC, 1998.

[44] Preston C., Random Fields, Springer, Berlin, 1976.

[45] Ruelle D., Superstable interactions in classical statistical mechanics, Communications in Mathematical Physics 18 (2) (1970) 127–159.

[46] Ripley B. D. and Kelly F. P., Markov point processes, Journal of the London Mathematical Society 2 (1) (1977) 188–192.

[47] Descombes X., Méthodes stochastiques en analyse d'image: des champs de markov aux processus ponctuels marqués, Ph.D. thesis, Université Nice Sophia Antipolis (2004).

[48] Strauss D. J., A model for clustering, Biometrika 62 (2) (1975) 467–475.

[49] Baddeley A. and Lieshout M. V., Stochastic geometry models in high-level vision, Journal of Applied Statistics 20 (5-6) (1993) 231–256.

[50] Geyer C., Likelihood inference for spatial point processes, in: Stochastic geometry, Routledge, 2019, pp. 79–140.

[51] Moller J., Lectures on random Voronoi tessellations, Vol. 87, Springer Science & Business Media, 2012.

[52] Boots B. and Shiode N., Recursive voronoi diagrams, Environment and Planning B: Planning and Design 30 (1) (2003) 113–124.

[53] Shamos M. I. and Hoey D., Closest-point problems, in: 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), IEEE, 1975, pp. 151–162.

[54] Metropolis N. and Ulam S., The monte carlo method, Journal of the American statistical association 44 (247) (1949) 335–341.

[55] Metropolis N., Rosenbluth A. W., Rosenbluth M. N., Teller A. H. and Teller E., Equation of state calculations by fast computing machines, The journal of chemical physics 21 (6) (1953) 1087–1092.

[56] Fontaine S., Mcmc adaptatifs à essais multiples, Master's thesis, Université de Montréal (2019).

[57] Casella G. and Robert C. P., Post-processing accept-reject samples: recycling and rescaling, Journal of Computational and Graphical Statistics 7 (2) (1998) 139–157.

[58] Meyn S. P. and Tweedie R. L., Markov chains and stochastic stability, Springer Science & Business Media, 2012.

[59] Robert C. P. and Casella G., Monte Carlo Statistical Methods, Vol. 2, Springer, 1999.

[60] Green P. J., Reversible jump markov chain monte carlo computation and bayesian model determination, Biometrika 82 (4) (1995) 711–732.

[61] Méliot P.-L., Chaînes de markov, théorie et applications, https://www.imo.universite-paris-saclay.fr/~pierre-loic.meliot/markov/markov.pdf.

[62] Chib S. and Greenberg E., Markov chain monte carlo simulation methods in econometrics, Econometric theory 12 (3) (1996) 409–431.

[63] Dobrushin R. L., Central limit theorem for nonstationary markov chains. i, Theory of Probability & Its Applications 1 (1) (1956) 65–80.

[64] Winkler G., Image analysis, random fields and Markov chain Monte Carlo methods: a mathematical introduction, Vol. 27, Springer Science & Business Media, 2012.

[65] Geyer C. J. and Møller J., Simulation procedures and likelihood inference for spatial point processes, Scandinavian journal of statistics (1994) 359–373.

[66] Tierney L., Markov chains for exploring posterior distributions, the Annals of Statistics (1994) 1701–1728.

[67] Geman S. and Geman D., Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on pattern analysis and machine intelligence (6) (1984) 721–741.

[68] Elmossaoui H., Oukid N. and Hannane F., Construction of computer experiment designs using marked point processes, Afrika Matematika 31 (2020) 917–928.

[69] Elmossaoui H., Contribution à la méthodologie de la recherche expérimentale, Ph.D. thesis, Université Saad Dahleb, Blida, Algérie (2020).

[70] Elmossaoui H., **Ait Ameur A.** and Oukid N., Improving computer experiment designs with two-type marked point processes, International Journal of Analysis and Applications 23 (2025) 82.

[71] **Ait Ameur A.** , Elmossaoui H. and Oukid N., Construction de plans d'expériences numériques à partir de processus stochastique ponctuel, in: Actes Colloque MOAD'22 Méthodes et Outils d'Aide à la Décision, Université de Béjaia, (Algérie), 2022.

[72] Elmossaoui H., **Ait Ameur A.** and Oukid N., Using marked point processes for computer experiment design, in: Actes Colloque MOAD'24 The Sixth International Colloquium on Methods and Tools for Decision, Mouloud Mammeri University of Tizi-Ouzou, Algeria, 2024.

[73] Elmossaoui H., Oukid N. and **Ait Ameur A.** , Experiment designs using nearest-neighbor markov point process, in: Symposium National sur les Mathématiques Innovantes: Rétrospective et Perspectives, Mostaganem, Algérie, 2024.

[74] Morris M. D. and Mitchell T. J., Exploratory designs for computational experiments, Journal of statistical planning and inference 43 (3) (1995) 381–402.

[75] Franco J., Planification d'expériences numériques en phase exploratoire pour la simulation des phénomènes complexes, Ph.D. thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne (2008).

[76] Shewry M. C. and Wynn H. P., Maximum entropy sampling, Journal of applied statistics 14 (2) (1987) 165–170.

[77] Elmossaoui H. and Oukid N., New computer experiment designs using continuum random cluster point process, International Journal of Analysis and Applications 21 (2023) 51–51.

[78] **Ait Ameur A.** , Elmossaoui H. and Oukid N., New computer experiment designs with area-interaction point processes, Mathematics 12 (15) (2024) 2397.

[79] **Ait Ameur A.** , Elmossaoui H. and Oukid N., Generation of numerical experiment designs based on point stochastic processes, in: Proceedings of the 2nd International Conference on Mathematics and Applications, Blida, Algeria, 2023.

[80] **Ait Ameur A.** , Elmossaoui H. and Oukid N., Exploration of the exponential convergence speed of the metropolis-hastings algorithm, in: IC-NMAA'24 First International Conference on Nonlinear Mathematical Analysis and Its Applications, Bordj Bou Arréridj University, Algeria, 2024, p. 250.

[81] Elmossaoui H., **Ait Ameur A.** and Oukid N., Comparative evaluation of computer experiment designs versus new computer experiment designs with area interactions, in: IC-NMAA'24 First International Conference on Nonlinear Mathematical Analysis and Its Applications, Bordj Bou Arréridj University, Algeria, 2024, p. 266.

[82] Baddeley A. J. and Turner R., Practical maximum pseudolikelihood for spatial point patterns, Advances in Applied Probability 30 (2) (1998) 273.

[83] Picard N., Bar-Hen A., Mortier F. and Chadœuf J., The multi-scale marked area-interaction point process: a model for the spatial pattern of trees, Scandinavian journal of statistics 36 (1) (2009) 23–41.

[84] Nightingale G. F., Illian J. B., King R. and Nightingale P., Area interaction point processes for bivariate point patterns in a bayesian context, Journal of Environmental Statistics 9 (2).

[85] Robert F., Théorèmes de perron-frobenius et stein-rosenberg booléens, Linear Algebra and its applications 19 (3) (1978) 237–250.

[86] Devroye L. and Lugosi G., Combinatorial methods in density estimation, Springer Science & Business Media, 2001.

[87] Yan D.-M., Wang W., Lévy B. and Liu Y., Efficient computation of clipped voronoi diagram for mesh generation, Computer-Aided Design 45 (4) (2013) 843–852.

[88] **Ait Ameur A.** , Elmossaoui H. and Oukid N., Modeling interaction point processes based on minimum distance, in: Actes Colloque MOAD'24 The Sixth International Colloquium

on Methods and Tools for Decision, Mouloud Mammeri University of Tizi-Ouzou, Algeria, 2024, p. 16.

[89] Elmossaoui H., Oukid N. and **Ait Ameur A.** , Innovative computer experiment designs utilizing a three-marked strauss point process, in: The first National Conference on Applied Mathematics, Statistics and Applications NCAMSA 2024 Conference, 2024, p. 109.