الجمهورية الجزائرية الديمقراطية الشعبية

Ministry of Higher Education and Scientific Research

Saad Dahlab University, Blida 1

FACULTY OF SCIENCES

DEPARTMENT OF MATHEMATICS

MASTER'S THESIS

Specialty: Mathematics

Option: Stochastic Modeling and Statistics

TITLE

Analysis and Comparison of Experimental Designs: A Synthesis Based on Optimality Criteria

---

Presented by : FERSAOUI Norhanne

Defended before the jury composed of:

Mr MASSIED M.	MAA	President
Mrs MESSAOUDI N.A	MCB	Examiner
Mr. ELMOSSAOUI H.	MCA	Supervisor

2024/2025

الملخص

هذه الرسالة تقدم دراسة معمّقة لمختلف أساليب تخطيط التجارب، مع التركيز على التصاميم التقليدية والرقمية على حد سواء. يتم إجراء مقارنة مفصّلة بين تصاميم التجارب استنادًا إلى معايير المثالية مثل التفاوت، والمسافات، والإنتروبيا. يُولى اهتمام خاص بتطبيق هذه التصاميم في التجارب العددية، لا سيما تلك المستندة إلى عمليات شتراوس (Strauss) وعمليات النقاط الموسومة، مما يوفّر نظرة شاملة على استخدامها في سياقات متنوعة.

الكلمات المفتاحية: تصاميم التجارب، التصاميم الرقمية، معايير المثالية، التصاميم العاملية، التصاميم الهامشية.

ABSTRACT

This thesis offers an in-depth exploration of both classical and computer-based experimental designs, emphasizing their evaluation based on optimality criteria. A detailed comparison of experimental designs is conducted based on optimality criteria such as discrepancy, distances, and entropy. Particular emphasis is placed on the application of these designs in numerical experiments, including those based on Strauss and marked point processes, providing a comprehensive overview of their use in diverse contexts.

Keywords: Experimental designs, computer designs, optimality criteria, factorial designs, marginal designs.

ACKNOWLEDGMENTS

First and foremost, I thank Allah Almighty for granting me the strength, patience, and determination necessary to accomplish this modest work.

I would like to express my sincere gratitude to **Mr. Hichem ELMOSSAOUI**, Senior Lecturer at Saad Dahlab University of Blida 1, my thesis supervisor, for his high-quality guidance and invaluable advice throughout this work. I am especially grateful for his availability, the relevance of his directions, and the constant interest he has shown in my work. Beyond his scientific guidance, I am deeply grateful for the supportive environment he fostered. I greatly admire his fighting spirit and, above all, his kindness.

I would also like to warmly thank PhD student **Ahmed AIT AMEUR** for his encouragement and insightful advice that accompanied me throughout this research.

I also wish to express my heartfelt thanks to the members of the jury for the honor they do me by accepting to evaluate this work.

My thanks also go to **Mr. Omar TAMI**, Head of the Mathematics Department at Saad Dahlab University of Blida 1, for the valuable support he provides us.

I take this opportunity to express my sincere appreciation to all the professors of the Mathematics Department who contributed to my education.

I would also like to thank the members of the Department's administrative office, in particular: **Mr. H. Hadj Allah**, **Ms. N. Djenas**, **Ms. W. Gougan**, and **Ms. S. Takarli**, for their support and availability.

Finally, I am deeply thankful to my family, who have always supported and encouraged me in all my decisions.

DEDICATION

I dedicate this work to my beloved mother, whose strength, sacrifices, and unwavering love have been my greatest source of motivation. Her constant care and encouragement have shaped every step of my journey.

To my father, for his support and faith in my abilities.

To my beloved grandfather, who raised me as his own and passed away in 2023, your love, strength, and wisdom shaped who I am. Your memory lives on in every step I take, and your guidance continues to inspire me.

To my grandmother, for her endless love and for being a pillar in my life.

To my brothers Chakib, Iyad, and Adem, and to my sisters Lina, Nermine, Sara, Arwa, and Maria, for their unwavering support, affection, and constant presence in my life.

To my aunt Wissam, who has always been like an elder sister to me, and to my uncle, whose supportive presence has always meant a great deal.

To my close friends, for their patience, humor, and reassuring words during moments of doubt.

To everyone who believed in me, thank you from the bottom of my heart.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The methodology of experimental research (Design of Experiments, DoE) is valuable for anyone conducting scientific research or industrial studies. Using experimental designs to empirically study a response function presents specific challenges for both statisticians and researchers. With limited prior knowledge of the response behavior, and generally only a small number of observations available relative to the number of parameters in their potential models, they must decide, before collecting any data, not only which models to use but also how to organize the experiments. Indeed, the quality of the statistical analysis is closely tied to the experimental design used to collect the data. Furthermore, the construction of experimental designs often requires combinatorial analysis.

To address industrial objectives, it is sometimes necessary to conduct a series of experiments to gather the missing information. The high cost of experimentation and the importance of decisions made based on its results mean that relying solely on the experimenters intuition is not advisable. A methodological approach is required one that reduces experimental cost while ensuring optimal organization of the trials.

The aim of the design of experiments methodology is to offer one or more strategies for addressing specific problems in experimental research. In our work, the general principles for constructing experimental designs are presented using the concept of the experimental space. While the geometric representation of experimental points is intuitive, it becomes limited as the dimensionality of the space increaseshence, the use of a matrix representation.

The wide variety of designs found in the literature stems from the absence of a single design that simultaneously satisfies all optimality criteria. Each design offers advantages with respect to certain criteria and limitations with respect to others. Thus, compromises must be made according to the specific objectives of each study.

In this context, the objective of our thesis is to propose a comprehensive synthesis of both classical and computer experimental designs, through a comparative study based on a selection of optimality criteria. This study aims to highlight the strengths of each design and guide researchers in making context-relevant and informed decisions. for their experimental studies.

The thesis is organized into four chapters:

- Chapter one introduces general concepts of experimental design: its history, purpose, basic terminology (response, factors, experimental space), as well as the mathematical and statistical tools required for modeling, estimation, and result analysis.

- Chapter two presents the main classical and computer experimental designs. It covers traditional designs (factorial, composite, Box-Behnken, Doehlert, and others.) as well as marginal designs (Latin hypercubes, orthogonal arrays, low-discrepancy sequences). A dedicated section also addresses designs arising from computer experiments, particularly those based on point processes such as Strauss, marked, clustered, and spatial interaction processes.

- Chapter three focuses on the optimality criteria used to evaluate experimental designs. It distinguishes between criteria applied to numerical designs (discrepancy, distances, entropy) and those used for classical designs (A-optimality, D-optimality, E-optimality, G-optimality, orthogonality), in order to highlight the strengths and limitations of each design type.

- Chapter four presents a detailed comparison of the various experimental designs discussed in the thesis. This analysis underscores the advantages and disadvantages of each design based on optimality criteria and specific user requirements.

Finally, the thesis concludes with a synthesis of the findings and recommendations for the optimal use of experimental designs.

CHAPTER 1

GENERALITIES OF EXPERIMENTAL DESIGNS

This chapter provides a synthesis of the key assumptions underlying the use of the experimental design methodology. Essential for any researcher conducting scientific investigations or industrial studies, this method applies across various disciplines whenever the goal is to analyze the relationship between a response variable y and influencing factors x_i. Its effective application requires adherence to strict mathematical principles and a rigorous methodological approach.

1.1 History

The methodology of experimental designs is not a new technique. It has been part of scientific progress since the early 20th century and is closely linked to the development of statistical methods. The systematic study of experimental design has evolved over time, shaping modern statistical techniques and optimization strategies. As early as the Middle Ages, Nicolas Oresme (1325-1382) recognized the importance of empirical methods in his writings [1]. Later, Francis Bacon (1561-1626), whose work influenced Descartes and Leibniz, became one of the precursors of the experimental method [2]. The formalization of experimental design began in the early 20th century with the pioneering work of Ronald A. Fisher. In the 1920s, Fisher introduced fundamental principles such as randomization, replication, blocking, and analysis of variance, which laid the groundwork for modern design of experiments and statistical inference[3].

During the mid-20th century, George E. P. Box and William G. Hunter introduced factorial designs, which allowed the simultaneous study of multiple factors and their interactions. Factorial designs became widely used in industrial experiments, particularly in the fields of agriculture, chemistry, and engineering [4].

Building on Fisher's legacy, notable statisticians such as Frank Yates, William Youden, William Cochran, Robin Plackett, and John Burman played crucial roles in promoting the application of experimental design techniques beyond agronomy. In the 1950s, Box and his collaborators extended Yates' ideas by developing fractional factorial designs at two levels [4]. However, the most transformative contribution came from Genichi Taguchi and Yuin Wu Masuyama, who introduced orthogonal arrays to simplify the construction of experimental designs for addressing a wide range of industrial problems. These influential tables were published in 1959 and 1961, significantly impacting quality improvement processes [5].

The field of DOE has continued to advance, with researchers developing experimental designs for mixture problems [6] incorporating block effects [7], applying nonlinear models [8], accounting for spatial correlations, and designing experiments for computer-based simulations [9]. These contributions have further diversified the applications of DOE across various scientific and industrial domains

1.2 Interest of the experimental design method

In experimental research, the goal is often to understand how an outcomesuch as crop yield, chemical production cost, or engine wearis influenced by various factors (1.3.2). Researchers measure this outcome while systematically varying the factors under controlled conditions. This enables the development of mathematical models describing the relationship between inputs and responses.

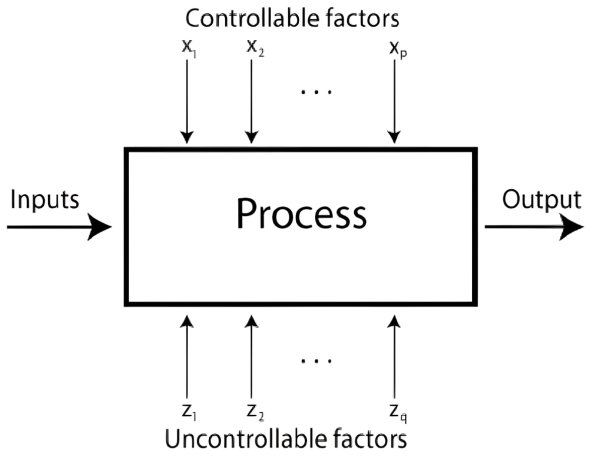

Figure 1.1: The system environment.

A key advantage of this method is the simultaneous variation of all factors in a structured

and systematic manner. Contrary to initial intuition, varying all variables at once is beneficial and offers several advantages, including:

- Reduction in the number of trials.
- Ability to study a large number of factors.
- Detection of interactions between factors.
- Improved precision of results.
- Modeling of results and determination of optimal conditions.

Understanding experimental designs relies on two essential concepts: the experimental space and the mathematical modeling of the studied quantities [10]. The experimental space represents all possible combinations of factor levels, guiding the planning of experiments. Mathematical modeling involves developing equations or algorithms that describe the relationship between factors and outcomes, enabling predictions and optimization.

By employing these strategies, researchers can efficiently explore complex systems, gain valuable insights, and make informed decisions based on empirical evidence.

1.3 Fundamental terminology of experimental designs

The Design of Experiments (DOE) methodology employs a specific terminology commonly used in experimental research. While these terms are widely recognized, their meanings can vary slightly across different statistical fields. To ensure clarity and consistency in this study, it is essential to define some key terms that will be frequently used throughout this work.

1.3.1 Response

The response is the dependent variable observed during the experiment. It reflects the effect of the studied factors and can be quantitative (e.g., yield, temperature) or qualitative (e.g., color, texture).

1.3.2 Factors and experimental space

Factors are variables that are studied for their potential influence on a system. The specific value assigned to a factor during an experiment is called a level. Factors can be classified into different categories:

- **Controllable factors**: These are variables that can be managed, adjusted, or modified during the experiment.
- **Non-controllable factors**: These factors are either considered negligible and kept at their usual values or are unknown influences that affect the experiment but cannot be controlled.
- **Quantitative factors**: These are expressed as measurable numerical values, such as speed, temperature, or intensity.
- **Qualitative factors**: These cannot be directly quantified; instead, they are represented by distinct categories, such as brand, process, method, or supplier.

When studying the effect of a factor, its variations are typically constrained within a defined range, with the low level (-1) representing the lower bound and the high level $(+1)$ representing the upper bound.

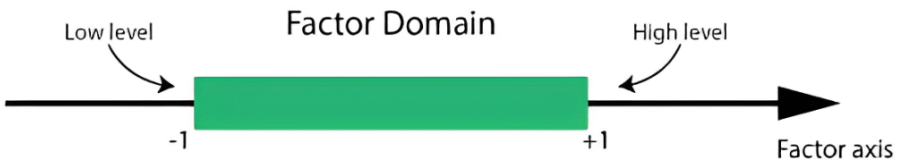

Figure 1.2: Factor variation range.

The effect of a factor refers to the variation in the response caused by a change in the factors level. The interaction between two factors represents the combined influence of both factors on the response, showing how the effect of one factor depends on the level of the other.

When introducing a second factor, it is represented by an additional axis. Like the first factor, it has a defined low level, high level, and range of variation. This second axis is positioned orthogonally to the first, forming a Cartesian coordinate system that defines a two-dimensional Euclidean space, known as the experimental space (Figure 1.3)

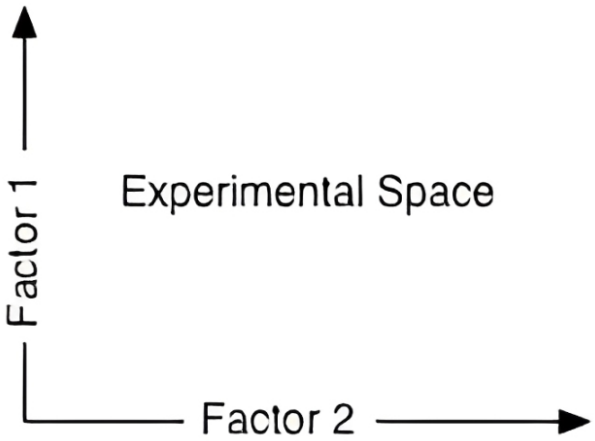

Figure 1.3: Experimental space definition.

The level X_1 of factor 1 and the level X_2 of factor 2 it's considerated as the coordinates of a point in the experimental space (Figure 1.4)

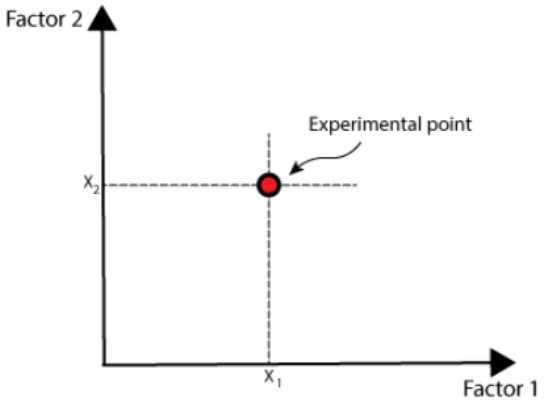

Figure 1.4: Experimental point in experimental space.

A given experiment is then represented by a point in this axis system, an experimental design is represented by a set of experimental points

1.3.3 Domain of study and Response surface

The study domain is defined by the combination of factor domains, representing the range of values that factors can take within an experiment. When considering k factors and their respective variations, the study domain forms a k-dimensional space, where each point corresponds to a unique configuration of the k factors. This space, also known as the research space, contains experimental points that can be positioned either inside or on the boundaries of the domain (Figure 1.5) [11]

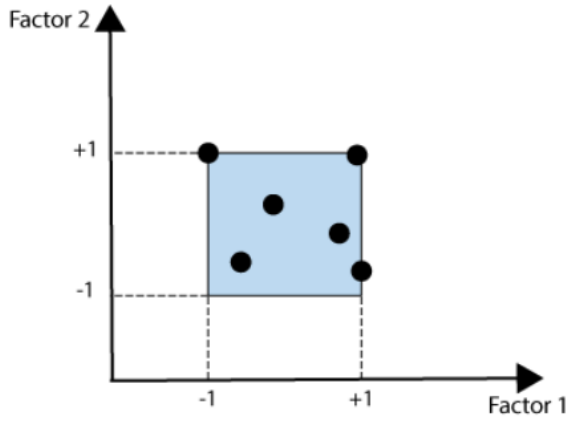

Figure 1.5: Two factors study domain

Each point within this study domain is associated with a response value, and the set of all responses forms a surface known as the response surface. Response surfaces can be classified into two categories:

- **Actual response surface**: Represents the real set of values taken by the response variable based on the process behavior.
- **Theoretical response surface**: When factors are continuous, an estimated response surface can be constructed using a mathematical model. In practice, this surface is derived from a limited number of experimental points, carefully selected by the experimenter (Figure 1.6).

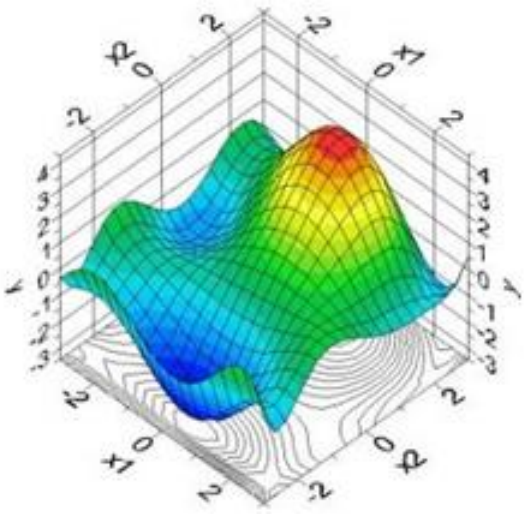

Figure 1.6: Definition of the response surface.

The fundamental challenge in experimental design is to determine an appropriate polynomial model that provides the best approximation of the actual response surface while minimizing

experimental effort. This approach is essential in optimizing processes and improving system performance [12].

1.3.4 Centered reduced coordinates

In experimental design, coding factor levels by assigning -1 to the low level and +1 to the high level introduces two key changes: Shift in Measurement Origin:

This adjustment centers the data around zero, facilitating easier interpretation of effects.

Change in Measurement Unit: Scaling the data standardizes the range of factor levels, allowing for uniform comparison across factors.

These transformations lead to the creation of centered and scaled variables, also known as coded variables. Centering refers to the change in origin, while scaling denotes the new unit of measurement. The transformation from the original variable z to the coded variable x is given by:

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

Here, z_0 represents the midpoint (average) of the high and low levels of z with:

$$z_0 = \frac{highlevel + lowlevel}{2}$$

and "step" is half the difference between these levels. with:

$$step = \frac{highlevel - lowlevel}{2}$$

This coding simplifies the design matrix, making it orthogonal and enhancing the interpretability of main effects and interactions. For example, in a full factorial design with three factors (A, B, and C), coding the factor levels as -1 and +1 allows for the systematic analysis of main effects and interactions using methods like Yates analysis. This approach exploits the structure of factorial designs to efficiently estimate factor effects.

1.3.5 Experimental Designs

Each point in the study area represents a possible operating condition, corresponding to an experiment that the operator can perform.

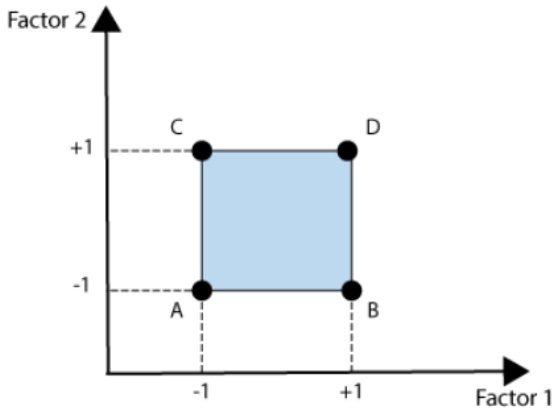

Figure 1.7: Corner Points A, B, C, and D

The fundamental challenge in experimental design lies in selecting the number and location of these experimental points. A set of experimental points that satisfies specific properties is referred to as an experimental design. Traditional experimental designs, which are well-established and extensively documented, fall under the category of classical designs. When experimental points are arranged in a manner deviating from these classical structures, they are classified as unconventional designs, often exhibiting inferior properties compared to classical ones [13].

1.3.6 Experimental Matrix

The experimental matrix shows all possible combinations of the low and high levels for each input factor. These high and low levels can be coded as -1 or +1. It is a table consisting of n rows, corresponding to the n experiments, and k columns, corresponding to the k variables (factors) being studied. The experimental matrix (Table 1.1) defines the trials represented in figure 1.7

runs	factor 1	factor 2
1(A)	-1	-1
2 (B)	+1	-1
3 (C)	-1	+1
4 (D)	+1	+1

Table 1.1: Experimental Matrix

1.4 Mathematical Tools for Experimental Design

In this section we will present the basic mathematical concepts necessary to understand the experimental design method. Mathematical modeling plays a crucial role in experimental design by providing a framework to describe and analyze the relationships between input factors and responses [11].

1.4.1 Concept of Mathematical Modeling

1.4.2 Statistical Model

A statistical model describes the relationship between input factors and responses, incorporating randomness and variability [14] Consider a random phenomenon dependent on k variables, where the objective is to model this phenomenon as accurately as possible. The statistical approach involves conducting n experiments, strategically chosen in the context of experimental design. Each experiment corresponds to a point x in $\mathbb{R}^k$ (assuming the variables are quantitative; for qualitative variables, a subset of $\mathbb{N}^k$ is used). The measured response, $Y(x)$, at point x is conventionally modeled as the sum of the true response function $f(x)$ (the actual response sought) and a residual term $\epsilon(x)$ (representing the experimental error)

A general form of a statistical model is: $Y(x) = f(x) + \epsilon(x)$

The residual can account for many causes such as errors due to the experimenter, a poor postulated model, the omission of certain variables. We generally assume that the residuals are real random variables satisfying the following three hypotheses [15]:

$$\begin{cases} \mathbb{E}(\epsilon(x)) = 0, & \forall x \\ \mathbb{C}ov(\epsilon(x), \epsilon(x')) = 0, & \forall x \neq x' \\ \mathbb{V}ar(\epsilon(x)) = \sigma^2, & \forall x \end{cases} \tag{1.1}$$

1.4.2.1 Linear Modeling

Linear modeling is widely used in experimental design to approximate relationships between variables In this section, we consider a statistical model that depends on k variables, where f is a linear function with respect to p unknown parameters. Mathematically, a model is linear in the parameters β_i $(i = 1, \ldots, p)$ if the partial derivatives $\frac{\partial f(x)}{\partial \beta_i}$ do not depend on β_i. Given a random phenomenon to be explained, it is generally not straightforward to propose

an appropriate model. The function f is often too complex, which is why it is common to approximate it using a set of standard functions (e.g., Taylor expansion, Fourier series, etc.).

If n experiments are conducted at points x_i $(i = 1, \ldots, n)$ in $\mathbb{R}^k$, we can express the response as:

$$Y(x_i) = f(x_i) + \varepsilon(x_i), \forall i = 1, \ldots, n \tag{1.4.1}$$

Since f is a linear function in terms of the unknown parameters, we can also write this model in matrix form as:

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

where:

- $Y \in \mathbb{R}^n$ is the vector of observed responses,
- $X(n, p)$ is the design matrix, which depends on the chosen experimental points and the assumed model,
- $\beta \in \mathbb{R}^p$ is the vector of unknown coefficients,
- $\varepsilon \in \mathbb{R}^n$ is the vector of residuals.

The assumptions (1.1) can be expressed as:

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

Consequently, $X\beta$ represents the expected (predicted) response given by the model.

1.4.3 Estimation of Coefficients Using the Least Squares Method

Once the model is established, the challenge lies in determining the best possible estimator $\hat{\beta}$ of β. A common approach is to find $\hat{\beta}$ such that the observed response vector Y and the predicted mean response vector $\hat{Y} = X\hat{\beta}$ are as close as possible.

Definition 1.1. The estimator $\hat{\beta}$ is called the least squares estimator of β if and only if $\hat{\beta}$ minimizes the objective function:

$$Q(\beta) = \|Y - X\hat{\beta}\|^2$$

The least squares estimator of β minimizes $Q(\beta)$, corresponding to the sum of squared errors between observed and predicted values:

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

This confirms that the quantity is directly related to the squared error between the observed responses Y_i and the predicted mean responses $\hat{Y}_i$. For the practical determination of this estimator, we have the following proposition:

Proposition 1.2. Given the statistical model $Y = X\beta + \varepsilon$ with X being a full-rank matrix[1], the least squares estimator of β is given by:

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

Proof: To find $\hat{\beta}$, we minimize the quantity:$\|Y - X\hat{\beta}\|^2 = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ Rewriting the sum in terms of $\hat{\beta}$

$$\begin{aligned}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 &= {}^t\left(Y - X\hat{\beta}\right)\left(Y - X\hat{\beta}\right)\\ &= \left({}^tY - {}^t\hat{\beta}\,{}^tX\right)\left(Y - X\hat{\beta}\right)\\ &= {}^tYY - {}^t\hat{\beta}\,{}^tXY - {}^tYX\hat{\beta} + {}^t\hat{\beta}\,{}^tXX\hat{\beta}.\end{aligned}$$

Note that $\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$ is a scalar, and it is easy to verify that all terms in the sum are also scalars. Therefore, we obtain:

$${}^tY - X\hat{\beta} = (\hat{\beta}^tX^tY)^t = \hat{\beta}^tX^tY$$

so

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 = {}^tYY - 2^t\hat{\beta}^tXY + {}^t\hat{\beta}\,{}^tXX\hat{\beta}.$$

To minimize the value of $\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$, we compute its derivative with respect to $\hat{\beta}$:

$$\frac{\partial\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\partial\hat{\beta}} = \frac{\partial^tYY}{\partial\hat{\beta}} - 2\frac{\partial^t\hat{\beta}^tXY}{\partial\hat{\beta}} + \frac{\partial^t\hat{\beta}^tXX\hat{\beta}}{\partial\hat{\beta}},$$

[1]A matrix is said to be full-rank if none of its columns are linearly dependent on the others, i.e., its rank is equal to the number of its columns.

where:

- $\frac{\partial\, {}^tYY}{\partial\hat{\beta}} = 0$, because ${}^tYY$ is a constant with respect to $\hat{\beta}$,
- $\frac{\partial\, {}^t\hat{\beta}{}^tXY}{\partial\hat{\beta}} = {}^tXY$, because ${}^t\hat{\beta}{}^tXY$ is a linear form with repect to $\hat{\beta}$,
- $\frac{\partial\, {}^t\hat{\beta}\, {}^tXX\hat{\beta}}{\partial\hat{\beta}} = {}^tXX\hat{\beta}$, because ${}^t\hat{\beta}\, {}^tXX\hat{\beta}$ is a quadratic form with respect to $\hat{\beta}$.

Thus:

$$\frac{\partial \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\partial\hat{\beta}} = -2{}^tXY + 2{}^tXX\hat{\beta}.$$

By setting this derivative to zero to find the minimum:

$$\frac{\partial \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\partial\hat{\beta}} = 0 \implies -2{}^tXY + 2{}^tXX\hat{\beta} = 0,$$

which leads to:

$${}^tXX\hat{\beta} = {}^tXY \implies \hat{\beta} = \left({}^tXX\right)^{-1}{}^tXY.$$

To verify that this value of $\hat{\beta}$ corresponds to a minimum, we compute the second derivative:

$$\frac{\partial^2 \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}{\partial\hat{\beta}^2} = 2{}^tXX.$$

Since X is full-rank, ${}^tXX$ is positive definite, meaning that the second derivative is strictly positive. Consequently, $\hat{\beta}$ is indeed a minimum.

Proposition 1.3. If the assumptions (1.2) on the residuals (errors) hold 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 the following variance-covariance matrix: $\mathbb{V}\left(\hat{\beta}\right) = \sigma^2\left({}^tXX\right)^{-1}$.

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

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

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

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

Expanding this expression:

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

Simplifying, we get:

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

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

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

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

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

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

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

Rearranging, we get:

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

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

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

Thus, by substitution:

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

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

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

Further simplification gives:

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

□

1.4.4 Prediction of the mean response

Once $\hat{\beta}$ has been estimated, the experimenter is often interested in using the obtained model to predict the mean response at a point where no experiment has been conducted. This prediction is crucial when the goal of modeling is, for instance, to determine the experimental conditions that maximize or minimize the studied response. The predicted mean response at a point $x \in \mathbb{R}^k$ is given by :

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

where $f(x) \in \mathbb{R}^p$ is a regression vector, constructed similarly to the rows of the matrix X Once the predicted mean response at x is determined, the accuracy of this prediction is assessed using the following result:

Proposition 1.4. The uncertainty associated with the prediction $\hat{Y}(x) =^t f(x)\hat{\beta}$ at $x \in \mathbb{R}^{\urcorner}$ is measured by

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

It can be observed that the error in the predicted response depends on four factors:

- The experimental error in the measured responses.
- The position of point x within the study domain.
- The set of points used to estimate the model coefficients, i.e., the experimental design itself.
- The assumed model used to interpret the results, through the coefficient computation 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 $\mathbb{V}(\hat{\beta}) = \sigma^2({}^tXX)^{-1}$*, the result follows.* □

1.4.5 Prediction variance function

The error associated with the measured responses depends on various factors, including the nature of the experimentation, the accuracy of the technology used, the care and skill of the experimenter, and other elements under their responsibility. These factors pertain to experimental practice rather than the theory of experimental designs [16]. To separate this experimental component from the theoretical one, 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 this variance function, we obtain the prediction error function:

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

1.4.6 Analysis of Variance (ANOVA)

Once the model is fitted, assessing the quality of the obtained fit becomes essential. This can be quantified using numerical indicators derived from *analysis of variance (ANOVA)* techniques. These methods rely on a structured decomposition of sums of squares to evaluate the model's explanatory power.

Let $\bar{Y}$ denote the observed mean response and Y^* the vector of centered observed responses. Notably, if $\mathbf{1_n}$ represents the unit vector of dimension n (i.e., a vector in $\mathbb{R}^n$ where all components are equal to 1), then [2]:

$$\bar{Y} = \frac{1}{n}^t \mathbf{1_n} Y, \quad Y^* = Y - \bar{Y}\mathbf{1_n}$$

We define the following three classical sums of squares (SS stands for Sum of Squares):

- **Total Sum of Squares (SST):**

$$SST = \sum_{i=1}^{n} (Y_i - \overline{Y})^2$$

- **Regression Sum of Squares (SSR):**

$$SSR = \sum_{i=1}^{n} (\hat{Y}_i - \overline{Y})^2$$

- **Error Sum of Squares (SSE):**

$$SSE = \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2$$

Proposition 1.5. For the least squares model, if $P = X({}^tXX)^{-1}\ {}^tX$ is the orthogonal projector onto $\mathrm{Im}(X)$ in $\mathbb{R}^n$, and if $I_n \subset \mathrm{Im}(X)$, then the sums of squares are given by:

$$\sum_{i=1}^{n} (Y_i - \bar{Y})^2 =^t YY - n\bar{Y}^2,$$
$$\sum_{i=1}^{n} (Y_i - \hat{Y})^2 =^t Y(I_n - P)Y,$$
$$\sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 =^t YPY - n\bar{Y}^2.$$

This leads to the fundamental decomposition:

$$\sum_{i=1}^{n} (Y_i - \bar{Y})^2 = \sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 + \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2.$$

Proof. In matrix form, we can write:

$$\sum_{i=1}^{n} (Y_i - \bar{Y})^2 =^t Y(I_n - \frac{1}{n} 1_n 1^t n)Y,$$

since $\bar{Y} = \frac{1}{n} 1^t nY$ and $1^t n 1_n = n$.

For the residual sum of squares:

$$\sum_{i=1}^{n} (Y_i - \hat{Y})^2 =^t Y(I_n - P)Y.$$

Since $\hat{Y} = PY$, we have:

$${}^tYY -^t \hat{Y}\hat{Y} =^t Y(I_n - P)Y.$$

For the regression sum of squares:

$$\sum_{i=1}^{n} (\hat{Y}_i - \bar{Y})^2 =^t YPY - n\bar{Y}^2.$$

Since $\hat{Y} = X({}^tXX)^{-1}\ {}^tXY$, multiplying by 1_n^T gives:

$$1_n^t \hat{Y} = 1_n^t Y.$$

Thus, we obtain:

$$\sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2 =^t Y P Y - n\bar{Y}^2.$$

For a random vector $Y \in \mathbb{R}^n$ and a non-random matrix $M \in \mathbb{R}^{n\times n}$, we define the degrees of freedom of ${}^tYMY$ as the rank of the matrix M. This concept arises from the chi-square distribution: if $Y \sim \mathcal{N}(\mu, \sigma^2 I_n)$ and M is a projection matrix, then ${}^tYMY$ follows a non-central chi-square distribution with a non-centrality parameter $\frac{1}{2}\ {}^t\mu A\mu$ and degrees of freedom equal to the rank of M [17]. □

1.5 Statistical Tests

1.5.1 The multiple correlation coefficient

The multiple correlation coefficient R^2 is a measure of how well a multiple linear regression model fits the data. It is defined 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)** represents the sum of squared residuals from the model, defined as:

$$SSE = \sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

- **SST (Total Sum of Squares)** is the total sum of squared differences between the observed dependent variable values and their mean:

$$SST = \sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

- **SSR (Sum of Squares due to Regression)** represents the sum of squares explained by the regres :

$$SSR = \sum_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$$

1.5.2 Fishers F-Test

Fishers test assesses the quality of the model fit. It is given by the following formula [18] :

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

where:

- $(p-1)$ is the degrees of freedom associated with SSR.
- $(n-p)$ is the degrees of freedom associated with SSE.

A high Fishers F-statistic indicates that the variance explained by the model is significantly larger than the residual variance, suggesting a good fit. To obtain statistically significant coefficients, F must be sufficiently large, corresponding to a low probability value.

CHAPTER 2

STUDY OF VARIOUS EXPERIMENTAL DESIGNS

In this chapter, we present the fundamental families of experimental designs without attempting to compare them. The various schemes are grouped into three complementary categories.

Standard designs are intended for estimating low-degree linear models. They include full and fractional factorial designs with two or three levels, as well as Mozzo designs. These are based on independent factors, meaning that each level can be freely set without imposing constraints on the others. Modeling designs are aimed at fitting quadratic responses or higher-order interactions. This category includes central and non-central composite designs, Box-Behnken designs, Doehlert designs, and Roquemore designs. As with standard designs, the factors remain independent, but the arrangement of points is primarily intended to enhance the accuracy of local approximations.

Finally, computer and space-filling designs are tailored for purely computational or highly expensive experiments. These include Latin hypercube designs, low-discrepancy sequences, space-filling models, and adaptive strategies. Their main purpose is to uniformly explore high-dimensional domains, often for use in metamodeling or sensitivity analysis.

Each section will describe the purpose, construction rules, and typical use cases of these design families, providing a clear overview of the tools available to practitioners.

2.1 Standard Designs

We have chosen to discuss standard designs in this thesis because they were initially developed for response surface applications. Among them, the most commonly used are factorial designs, Box-Behnken designs, and central composite designs, which are relatively easy to generate. We

have also chosen to include Doehlert designs, which are particularly well suited for space-filling considerations.

In this section, we first provide a brief description of these designs before analyzing the key properties of interest, such as space-filling capability, non-redundancy, and cost. It is worth noting that these designs will be frequently referenced throughout this thesis to assess the relevance of Space-Filling Designs.

2.1.1 Full factorial designs

The simplest method to achieve proper space-filling is to select points on a regular grid within the experimental domain.

Description: To construct a regular grid with k levels, one simply needs to choose k values evenly spaced across the range of each factor. For example, in the unit square $[0,1]\times[0,1]$, selecting 5 levels results in the following grid of points (see Figure 2.1)

$$\{0, 0.25, 0.5, 0.75, 1\}\times\{0, 0.25, 0.5, 0.75, 1\}$$

:

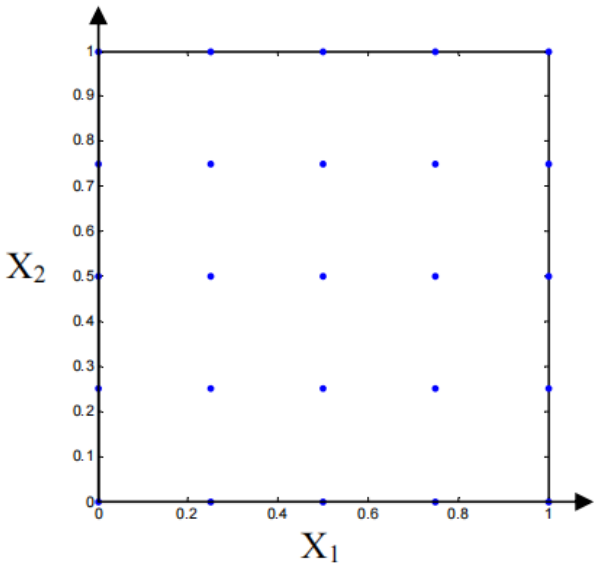

Figure 2.1: A full factorial design with 5 levels.

It is evident that the higher the number of levels, the better the space-filling quality. However, this also leads to an exponential increase in the number of simulations. Therefore, it is essential to find a suitable trade-off by selecting the most relevant levels for the problem at hand.

Discussion: This method remains effective when the problem's dimensionality is low, typically limited to 2 or 3 variables. However, as the dimensionality increases, the number of simulations n^d grows exponentially, making grid-based approaches impractical. Moreover, each dimension only takes k distinct values. If the response depends primarily on a few variables (e.g., one or two in a five-dimensional space), a factorial design results in many redundant points. Consequently, this type of design becomes inefficient in high-dimensional settings, as most points are lost when projected onto the factorial axes. For instance, if the response follows the form $f(X_1, X_2) = f_1(X_1)$ or $f(X_1, X_2) = f_2(X_2)$, then the factorial design illustrated in Figure 2 is poorly suited, as it effectively reduces the available information to only 5 points instead of 25.

Remark 1. : If the number of model coefficients to be estimated is close to the number of experimental runs, it is advisable to enhance the factorial design by adding a few points uniformly distributed within the experimental domain.

2.1.2 Fractional factorial designs

Given the constraints that prevent us from conducting a large number of simulations, *full factorial designs* are not suitable. However, the underlying principle remains valuable. Therefore, fractional factorial designs present a good alternative. By selecting subsets of full factorial designs, the number of required simulations can be significantly reduced, leading to lower experimental costs (for more details, see Myers & Montgomery, 1995).

However, the issues related to factorial projections remain present, as observed with full factorial designs. Additionally, new alignment problems arise due to aliasing effects inherent in fractional designs, similar to what occurs in orthogonal linear arrays.

2.1.3 Composite Designs

A composite experimental matrix is a combination of:

- A two-level factorial design matrix, which can be either full factorial (2^d) or fractional factorial (2^{d-r}), where the points correspond to the vertices of a hypercube (e.g., $[-1, 1]$).
- An axial design matrix, consisting of points symmetrically placed along each axis at a distance [1] α from the center of the domain.

[1] For a cubic domain, the value of α is typically set to 1.

- A central point [2], which, for d factors, provides information about the variability of the phenomenon and allows for testing the models validity. For example, in the case of a first-degree linear model, it helps detect the presence of curvature.

A face-centered composite design within the cubic domain $[-1, 1]^2$ corresponds to a three-level factorial design $(-1, 0, 1)$. Notably, these designs are well-suited for the one-at-a-time (OAT) approach, as they impose points along the axes and within the factorial design 2^d

Composite designs are widely used in classical experimentation to approximate second-degree response surfaces [19]. Different types of composite designs can be generated by adjusting the distance between the central point and the boundary points of the domain. Common examples include:

- **Central composite designs (CCD)**
- **Face-centered composite designs**
- **Inscribed central composite designs**

However, the number of experiments in composite designs increases rapidly with the number of factors, primarily due to the factorial matrix. These designs do not optimally fill the experimental space and often fail to achieve good point distribution in projections. Indeed, they test only three or five levels per parameter (depending on α, regardless of the design size.

2.1.4 Box-Behnken Designs

Box-Behnken designs are experimental designs where variables take only three levels $(-\alpha, 0, +\alpha)$, considering the experimental domain as a hypercube $[-1, +1]^d$ These designs consist of:

- A two-level factorial matrix (2^d points).
- Balanced incomplete blocks, arranged in a specific pattern.
- central point, added to the matrix to improve estimation accuracy.

Box-Behnken designs serve as an alternative to composite designs since they require only three levels per factor [20] while still allowing for the modeling of a second-degree response surface. The construction methods for these designs, including the specific way to form the blocks, can be found in [19] and [21]

[2] In classical experimentation, it is recommended to include multiple central points to assess experimental variability. However, this approach becomes irrelevant when using a purely deterministic simulator, as there is no inherent variability in the response.

In terms of the number of experimental points, a **Box-Behnken design** is comparable to a composite design in dimensions 3 and 4. However, there is no Box-Behnken design for two factors. Because these designs place their points on the factorial axes rather than throughout the domain, they do not ensure a good space-filling property.

2.1.5 Doehlert Designs or Uniform Networks

Doehlert designs (Doehlert, 1970) belong to the family of uniform networks. Their generation method is iterative and consists of:

- Defining an initial simplex within the exploration domain.
- Applying isometries (translations and rotations) from one of its vertices (typically through translations).
- Iterating this process, which results in a specific distribution of points (as shown in Figure 2.2).

Practically, for each variable in the range $[-1, 1]$, this approach involves successively subtracting the coordinates of each point in the initial simplex from the others.

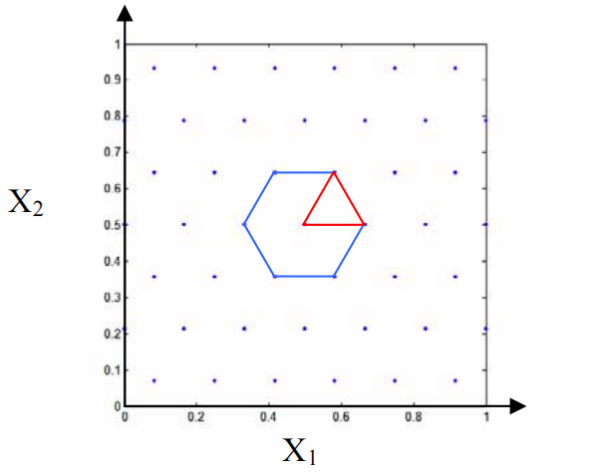

Figure 2.2: A Doehlert design with 45 points in the unit square and its initial simplex: the equilateral triangle in red.

Example 1. Figure 2.2 illustrates a Doehlert design with 45 points within the unit square, where the initial simplex is an equilateral triangle (highlighted in red).

2.1.6 MOZZO Designs

MOZZO designs are characterized by their sequential nature [22]. Initially, two factors can be studied using three experimental runs within a triangular domain. If the decision is made to include a third factor, three additional runs are carried out (runs 4, 5, and 6 in Table 2.1). This sequential structure is only possible if the factors not yet under investigation are held constant during the study of the initial factors.

For example, Factor 3 is fixed at level -1 while Factors 1 and 2 are being studied. To investigate Factor 3, its level is changed to +1, and another triangular design is executed with the first two factors. As more factors are introduced, corresponding interactions can be incrementally added to the initial base model.

Trial No	Factor 1	Factor 2	Factor 3	Factor 4
1	0,268	1	-1	-1
2	0,732	-0,732	-1	-1
3	-1	-0,268	-1	-1
4	-0,268	-1	1	-1
5	-0,732	0,732	1	-1
6	1	0,268	1	-1
7	-0,267	-1	- 1	1
8	-0,732	0,732	-1	1
9	-1	0,268	-1	1
10	0,268	-1	1	1
11	0,732	-0,732	1	1
12	1	0,268	1	1

Table 2.1: Mozzo Experimental Designs for Two, Three, and Four Factors

2.1.6.1 Mozzo Design for Two Factors

This design allows for the study of two factors using only three experimental runs, arranged in a triangular configuration. Figure 2.3 illustrates one possible configuration of this triangular layout.

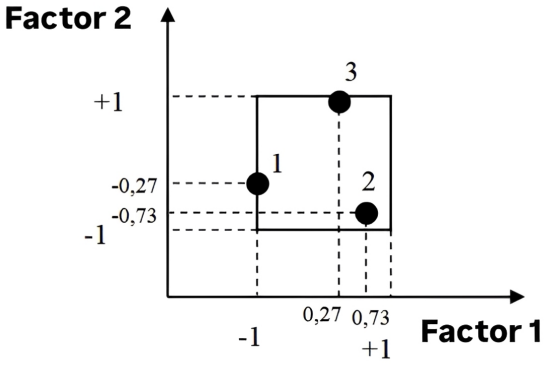

Figure 2.3: Study Domain of the Mozzo Design for Two Factors

Given the small number of points, the assumed mathematical model is simple: a first-order model without interaction terms, expressed as:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3$$

We now write the corresponding design matrix $\mathbf{X}$ for the model.

$$X = \begin{pmatrix} 1 & 0,268 & 1 \\ 1 & 0,732 & -0,732 \\ 1 & -1 & -0,268 \end{pmatrix}$$

The information matrix $\mathbf{X}\mathbf{X}^T$ is computed directly and yields:

$$\mathbf{X}\mathbf{X}^T = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 1,608 & 0 \\ 0 & 0 & 1,608 \end{pmatrix}$$

This result confirms that the design matrix $\mathbf{X}$ is orthogonal. Furthermore, the elements corresponding to the first-order terms are equal. This implies that the Mozzo design satisfies the iso-variance by rotation criterion 3.2.8, ensuring that the prediction error remains constant for all directions equidistant from the center of the experimental space.

2.1.6.2 Mozzo Design for Three Factors

This configuration corresponds to the first six runs of Table 2.1. Figure 2.4 illustrates the spatial distribution of the experimental points in the three-dimensional experimental space.

The design is sequential and symmetrical, allowing for the progressive inclusion of additional factors while preserving the geometric balance of the design.

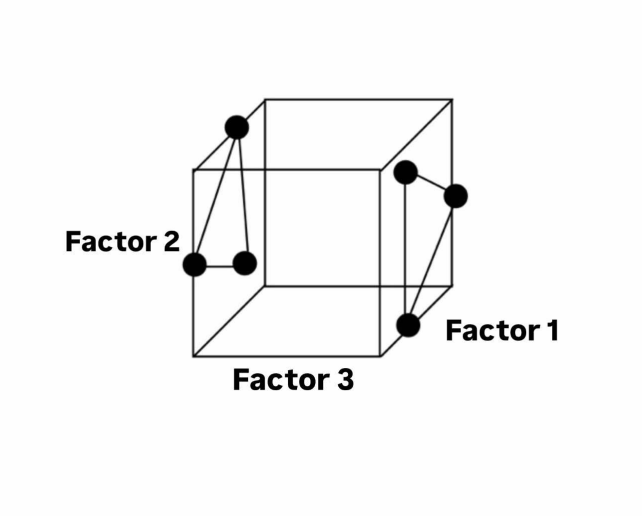

Figure 2.4: Study Domain of the Mozzo Design for Three Factors

As there are six experimental points, it is theoretically possible to estimate six unknown parameters. Therefore, a first-degree model with interactions can be considered. However, due to the configuration of the experimental points, it is not possible to estimate interactions involving the third factor. Only the interaction between factors 1 and 2 can be included in the model. The resulting model is given by:

$$y = a_0 + a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2$$

The corresponding design matrix X is:

$$X = \begin{pmatrix} 1 & 0.268 & 1 & -1 & 0.268 \\ 1 & 0.732 & -0.732 & -1 & -0.536 \\ 1 & -1 & -0.268 & -1 & 0.268 \\ 1 & 0.268 & -1 & 1 & 0.268 \\ 1 & 0.732 & 0.732 & 1 & -0.536 \\ 1 & 1 & 0.268 & 1 & 0.268 \end{pmatrix}$$

To verify orthogonality, we compute the information matrix $X^\top X$:

$$X^\top X = \begin{pmatrix} 6 & & & & \\ & 3.22 & & & \\ & & 3.22 & & \\ & & & 6 & \\ & & & & 0.86 \end{pmatrix}$$

(Note: The full calculation of $X^\top X$ is omitted here but should be completed if required.)

This computation allows us to verify whether the orthogonality property is preserved in the presence of the added interaction term.

2.1.6.3 Advantages and Limitations

- **Main advantage:** The primary benefit of Mozzo designs lies in the very limited number of required experimental runs. For two factors, only three experiments are needed, and each factor is tested at three different levels.
- **Limitations:** Mozzo designs are not available for every possible number of factors. Additionally, the proposed model generally does not account for all possible interactions between the factors.

2.2 Marginal designs

In this section, we introduce designs that, by construction, exhibit good properties in terms of non-redundancy and non-alignment with certain subspaces. However, there is no guarantee that they effectively cover the experimental space, which we will investigate here.

We also define the concept of margins, which refers to factorial subspaces. For instance, one-dimensional margins correspond to factorial axes.

2.2.1 Latin Hypercubes

The Latin hypercube sampling method, introduced by MacKay, Conover, and Beckman in 1979, was developed for the numerical evaluation of multiple integrals. It ensures non-redundant information through well-distributed projections on factorial axes. In practice, Latin hypercubes are widely used in numerical experimental design, particularly due to their ease of implementation and construction. Description Each axis $[0, 1$ of the unit cube is divided into nn segments of equal length according to the following subdivision: $[0, \frac{1}{n}], [\frac{1}{n}, \frac{2}{n}], \ldots, [\frac{n-1}{n}, 1]$

By taking the Cartesian product of these intervals, we obtain a grid of ndnd cells of equal size. Then, nn cells are selected among the ndnd possible ones in such a way that each one-dimensional margin is represented exactly once. Finally, a random point is drawn within each of the preselected cells.

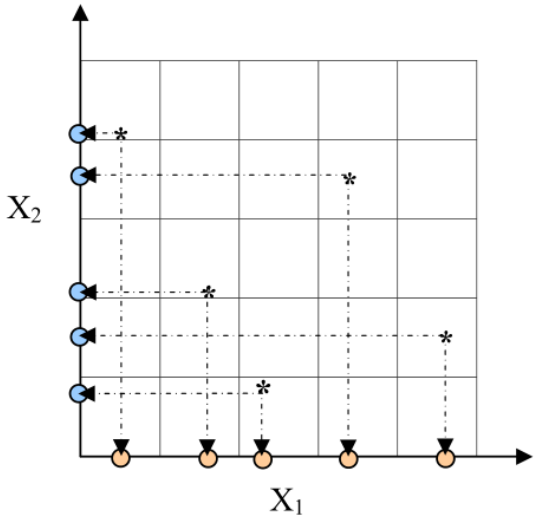

Figure 2.5: A Latin hypercube sampling with 5 points in 2 dimensions.

Definition 2.1. A Latin hypercube with nn points in $[0,1]^d$ is defined as the set of points X^i such that:

$$X_j^i = \frac{\pi_j(i) + U_j^{(i)}}{n}, 1 \leq i \leq n, 1 \leq j \leq d$$

where π_j is a permutation of $1, ..., n$, and $U_j^{(i)}$ $U[0,1]$ is a random variable following a uniform distribution on $[0,1]$.

Thus, the vector $(\pi_1(j), \ldots, \pi_d(i))$ represents the cell in which the point X^i is located, while $(U_1^i, \ldots, U_d^i)$ determines its exact position within the cell.

The resulting Latin hypercube can be represented as a matrix with n rows and d columns, whose coefficients are X_j^i.

Remark 2.

1. Points can be placed at the center of the cells to eliminate randomness in the design.

2. A Latin hypercube, defined by the matrix π, is very easy to construct since each column is a permutation of $1, ..., n$.

Discussion

Latin hypercube points have the interesting property of being uniformly distributed along factorial axes (see Figure 2.5). However, this property does not necessarily ensure uniformity across the entire experimental domain.

For a fixed n, there are $n!$ possible permutations for each of the d columns, leading to a total of $(n!)^{(d-1)}$ possible Latin hypercubes. However, not all of them ensure a uniform distribution of points in the space.

For instance, in the Latin hypercube shown in Figure 2.6, the points are aligned along one of the domains diagonals. If the actual process depends only on X_2X_1, then the information provided by this experimental design is reduced to a single point instead of five, significantly limiting the sampling quality.

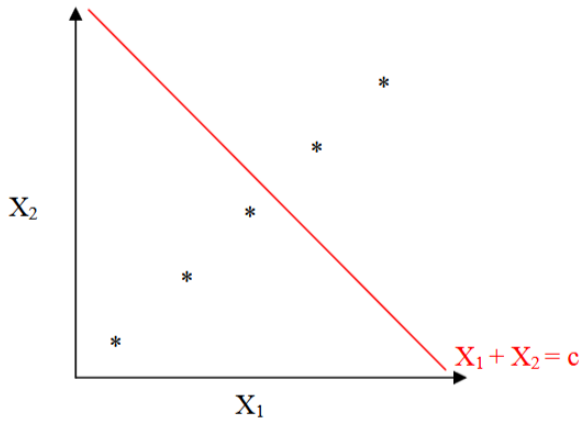

Figure 2.6: A Latin hypercube sampling with 5 points in dimension 2.

2.2.2 Orthogonal Arrays

2.2.2.1 General Case

Conceptually, orthogonal arrays [23] are very similar to Latin hypercubes. Indeed, they share the advantageous high-dimensional projection properties of Latin hypercubes in one dimension.

Definition 2.2. An orthogonal array of strength t with q symbols is a matrix with n rows and d columns (where $d > td > t$), whose elements take q distinct values. This matrix is structured so that every submatrix of size $n \times t$ contains each possible combination of t symbols exactly λ times.

Thus, the following relation holds:

$$n = \lambda q^t$$

Such an orthogonal array is denoted as $OA(n, d, q, t, \lambda)$.

Description From a geometric perspective, this corresponds to subdividing the unit cube axes into q equal segments, resulting in q^d equally sized cells. Then, n cells are selected to form an orthogonal array that meets the above definition.

This structure ensures that every set of t columns in the design matrixi.e., each t-tuple of symbolsappears exactly λ times.

Remark 3. n orthogonal array of strength 1 is equivalent to a Latin hypercube. As with Latin hypercubes, the sampling point can be chosen randomly within each cell or placed at the center. In the latter case, all projections onto tt-dimensional subspaces result in a regular grid, as illustrated in Figure

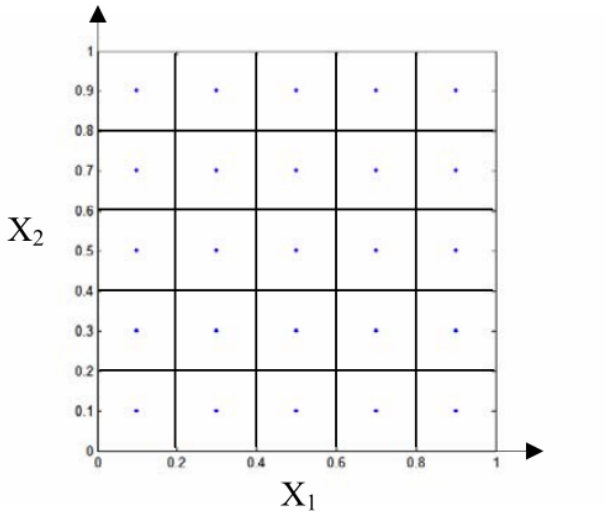

Figure 2.7: A design generated by an orthogonal array $OA_1(25, 5, 5, 2)$, with points centered and projected onto the (X_1, X_2) subspace.

Definition 2.3. An orthogonal array sampling (hereafter referred to as an orthogonal array) with nn points in $[0, 1]^d$ is a set of points X^i defined by:

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

where:

- π_j is a permutation of $\{0, ..., q-1\}$.
- A_j^i are the elements of the orthogonal array.
- U_j^i $U(0, 1)$ is a uniformly distributed random variable in $[0, 1]$.

Thus, the vector $(\pi(A_1^i), \pi(A_2^i), \ldots, \pi(A_d^i))$ represents the cell where point X^i is located, while $(U_1^i, \ldots, U_d^i)$ defines its relative position within that cell.

The orthogonal array corresponds to a matrix with n rows and d columns, where each coefficient is given by X_j^i.

Property 2.4. The generation of these designs follows a property similar to that of Latin hypercubes: If the symbols in each column of an orthogonal array of strength t are permuted, the resulting array remains an orthogonal array of strength t.

2.2.2.2 Special Case of Linear Orthogonal Arrays

Linear orthogonal arrays form a specific subclass of orthogonal arrays, chosen for their ease of implementation compared to the general case.

Definition 2.5. A linear orthogonal array is an orthogonal array that satisfies the following conditions:

- The number of symbols q is a prime number.
- The rows of the array are all distinct and constitute a vector subspace of $\mathbb{Z}_q^d$

In this case, the array is denoted as: $OA(d, q, t, \lambda)$ over $\mathbb{Z}_q$ where $\mathbb{Z}_q = 0, ..., q-1$ forms a finite field since q is a prime number.

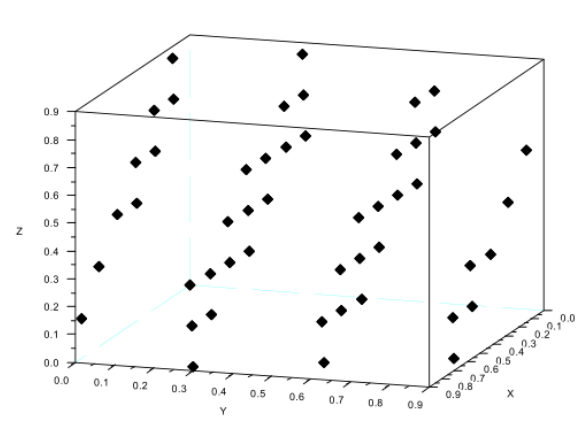

Figure 2.8: A distribution of 49 points derived from a linear orthogonal array of strength 2 in dimension 3.

Remark 4. A linear orthogonal array of strength t is simply an orthogonal array of strength t structured as a vector subspace. In particular, a linear orthogonal array of strength 1 is always a Latin hypercube.

Regarding the construction of linear orthogonal arrays, readers may refer to Jourdan's (2000) [24] dissertation. However, upon analyzing the designs generated using this method, we observed additional alignment issues compared to traditional orthogonal arrays. Specifically, the points are distributed along parallel planes (see Figure 2.8).

This phenomenon arises because the computations are performed in $\mathbb{Z}_q$. It is even possible to determine the equation of the planes on which the points are distributed. For instance, in the case of a linear orthogonal array of strength 2 over $\mathbb{Z}_7$, the construction method ensures that the points satisfy the following condition: $x + y + z = 0 \ (mod\ 7)$

As a result, the points are located on five parallel planes (four of which are clearly visible in Figure 2.8, while the fifth consists only of the origin). Consequently, the distribution of projections along the axis perpendicular to these planes is not optimal.

2.2.3 Latin Hypercubes Based on Orthogonal Arrays of Strength 2

Orthogonal arrays are widely used in experimental design, mainly due to their favorable uniformity properties. However, for orthogonal arrays of strength $t > 1$, this uniformity is only guaranteed on subspaces of dimension t. As a result, these arrays exhibit repetitions along factorial axes.

Thus, using Latin hypercubes seems like a promising alternative to ensure a better representation of factorial axes. However, these designs do not necessarily provide a uniform distribution across subspaces of dimension $t > 1$.

To summarize, neither method is entirely satisfactory.

To address these issues, Tang 1993 [25] proposed an approach that combines:

- the orthogonality properties of orthogonal arrays,
- the favorable projection properties of Latin hypercubes.

Tang also introduced an algorithm to generate these designs from orthogonal arrays of strength 2, ensuring good uniformity on 1-dimensional margins.

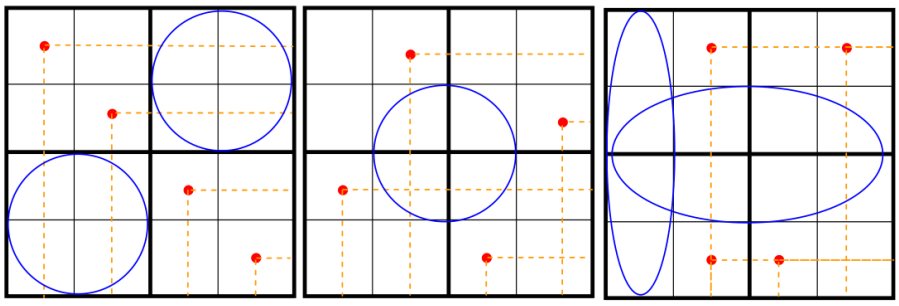

Figure 2.9: A Latin hypercube, a Tang Latin hypercube, and an orthogonal array of strength 2 with 4 points in dimension 2.

Definition 2.6. Let A be an orthogonal array of type $OA(n, d, q, 2)$. For each column of A,

we replace each element by a permutation of the set of q elements according to the following rule:

$\forall k, k \in \{0, 1, \ldots, q-1\}$, $[kq+1, kq+2, \ldots, (k+1)q]$

This transformation results in a Latin hypercube.

for example if $A = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix}^t$ and we apply the following permutations for each column of A:0→0 then 0→1 and 1→3 then 1→2

we then obtain the following Latin hypercube: $\begin{pmatrix} 0 & 1 & 3 & 2 \\ 0 & 3 & 1 & 2 \end{pmatrix}^t$

Conversely, starting from a Latin hypercube, it is possible to reconstruct an orthogonal array of type $OA(n, d, q, 2)$, where the coefficients are defined as follows:

$X_{ij} = \frac{X_{ij}}{q}$,for $i = 1, ..., n$ and $j = 1, ..., d$ where denotes the floor function.

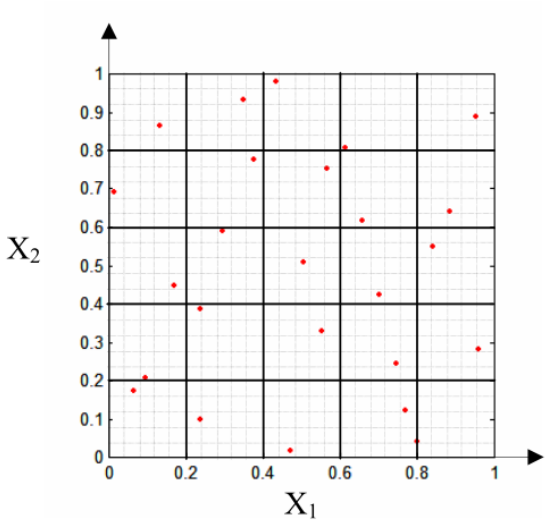

Figure 2.10: A Latin hypercube generated from an orthogonal array $OA_1(25, 5, 5, 2)$ with randomized points projected onto the subspace (X_1, X_2).

Description This method follows a three-step sampling process.

First, the unit cube is divided into q^d cells. Then, among these cells, $n = q^2$ are selected in such a way that they form an orthogonal array of strength 2.

Second, a sub-cell is chosen within each of the n selected cells, ensuring the construction of a Latin hypercube.

Finally, a random point is placed within each sub-cell, resulting in a Latin hypercube based on an orthogonal array (see Figure 2.10).

2.2.4 Low-discrepancy sequences.

In the previous section, we discussed designs where points are well distributed in projection but not necessarily evenly spread in space. Here, we introduce designs aimed at achieving a more uniform filling of the space while also examining their projection properties.

These point sequences were originally developed to replace random sequences in Monte Carlo methods, leading to the term quasi-Monte Carlo methods. Most *low-discrepancy sequences* [26] are generated using deterministic algorithms to ensure that points are distributed as uniformly as possible within the experimental domain.

To provide a fundamental understanding of how these sequences achieve space-filling properties, we introduce a basic definition of discrepancy. Niederrieters (1987) definition, presented in Section 2.1, offers deeper insight into the theoretical foundation of discrepancy. Readers may refer to that section for various methods of computing discrepancy.

Discrepancy measures the deviation between a given point distribution and a perfectly uniform distribution; in other words, it quantifies the irregularity of point dispersion. In the one-dimensional case, given the empirical distribution function $\hat{F}_n$ of the points $x_0, x_1, ..., x_{n-1}$, discrepancy is defined as:

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

where $F_U(x)$ is the cumulative distribution function of the uniform distribution on $[0, 1]$.

Remark: The function $D_n(X)$ corresponds to the Kolmogorov-Smirnov statistic, which is commonly used to test the goodness-of-fit to a uniform distribution.

Definition 2.7. Uniform Distribution: Let X be a compact space and μ a regular probability measure defined on the Borel sigma-algebra of X. A sequence of points $(x_n)_{n\in\mathbb{N}}$ in X is said to be **uniformly distributed** if, for any continuous function $f \in C(X)$, we have:

$$\frac{1}{n}\sum_{k=1}^{n} f(x_k) \to \int_X f\, d\mu, \quad \text{as } n \to \infty.$$

Remark 5. The strong law of large numbers ensures this convergence almost surely.

An important property in this context is the following.

Property 2.8. A sequence (x_n) is uniformly distributed if

$$\lim_{n\to\infty} D_n(x) = 0. \tag{2.2.1}$$

There exist numerous upper bounds on discrepancy. The most well-known result in this regard is the Koksma-Hlawka inequality.

Theorem 2.2.1. *If f is a function of bounded variation $V(f)$ in the sense of Hardy and Krause, then for any sequence of points $x_1, ..., x_n$ in $[0,1]^d$, we have:*

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(x_i) - \int_{[0,1]^d} f(t)dt \right| \leq V(f) D_n(X). \tag{2.2.2}$$

Thus, the worst-case approximation error is the product of the variation $V(f)$ (which reflects only the irregularity of the function f) and the discrepancy $D_n(X)$ (which measures only the quality of the sequence's distribution).

Remark 6. The sequences discussed here will be finite and conventionally indexed from 0 to $n-1$ to include the origin of the domain.

It is possible to construct sequences whose discrepancy is lower than that of a random sequence, which is of order $\frac{1}{n}$. These are known as low-discrepancy sequences. Such sequences are characterized by their ability to fill the unit cube uniformly and with an extremely regular pattern.

A natural approach to achieving the most uniform distribution of points is to consider a regular grid. However, it can be shown that the discrepancy of such a distribution remains of order $\frac{1}{n}$, which is actually a poor result. The reasons behind this will be further discussed in Section 3.1 , dedicated to discrepancy computation.

Furthermore, we will see that the complexity of discrepancy computation depends on the dimensionality, making it impractical for high-dimensional cases. This is why the low-discrepancy sequences introduced here are particularly useful, as they are easy to implement and ensure low discrepancy.

Examples of low-discrepancy sequences were proposed by Halton [27], Hammersley [28], Sobol [29], Faure [30], and Niederreiter [31]. In the following sections, we will study the construction and properties of these different sequences used in experimental design.

A fundamental concept behind the construction of most of these sequences is the inverse radical function in base b, defined as follows:

All the sequences introduced below are defined for all n. We will see that most of these sequences are valuable due to their iterative properties for example, when adding q points to an existing design of size n. This property significantly influences the choice of the sequence to be used.

Definition 2.9. Let b be an integer with $b \geq 2$. The inverse radical function in base b is given by:

$$\phi_b(i) = \sum_{m=0}^{\infty} \frac{p_m}{b^{m+1}}$$

where i is represented in base b as:

$$i = p_0 + p_1 b + p_2 b^2 + \cdots + p_m b^m$$

with p_m being the digits of i in base b. The sequence: $C_b = \{x_0, x_1, \ldots, x_{n-1}\}$, where $x_i = \phi_b(i)$ is called the Van Corput sequence in base b [3].

2.2.4.1 Halton Sequences

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

Definition 2.10. A Halton sequences $H_{b_1,\ldots,b_d} = \{x_0, x_1, \ldots, x_{n-1}\}$ in bases $b_1, \ldots, b_{n-1}$ is defined as:

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

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

Remark 7. To minimize the discrepancy, it is recommended to choose the first d prime numbers as bases. This choice helps reduce the leading term constant in the upper bound of the discrepancy of such sequences (see Faure[32], revisited in Niederreiter [33]).

Halton sequences have the advantage of being easy to implement and computationally efficient. Transitioning from $x^i = \phi_b(i)$ to $x^{i+1} = \phi_b(i+1)$ simply requires an addition in base b, making them well-suited for practical applications.

[3] The Van Corput sequence is a low-discrepancy sequence used in quasi-random sampling; it distributes points uniformly over the interval [0, 1] by reversing the digits of natural numbers in a given base.

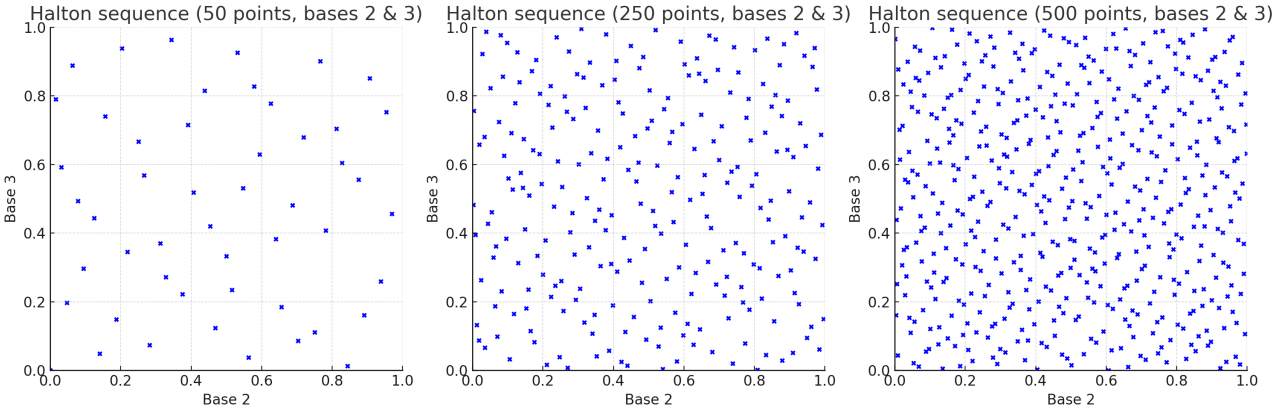

Figure 2.11: The first 50, 250, and 500 points of a Halton sequence in bases 2 and 3.

2.2.4.2 Hammersley Sequences

A Hammersley sequence in dimension d is constructed using a term dependent on the number of points and a Halton sequence in dimension $d-1$.

Definition 2.11. A Hammersley sequences $H^n_{b_1,\ldots,b_{d-1}} = \{x_0, x_1, \ldots, x_{n-1}\}$ in beses $b, \ldots, b_{d-1}$ is defined by

$$x_i = (\frac{i}{n}, \phi_{b_1}(i), \ldots, \phi_{b_{d-1}}(i)) \quad \in [0,1]^d$$

where $b_1, \ldots, b_{d-1}$ are pairwise coprime positive integers.

Remark 8. To minimize the discrepancy of $H^n_{b_1,\ldots,b_{d-1}}$, it is recommended to choose the first $d-1$ prime numbers as bases.

Discussion Since Hammersley sequences are built from Halton sequences, they exhibit the same pattern of successive diagonals. Moreover, it is not possible to add extra points to a Hammersley sequence without affecting its discrepancy. Therefore, when the number of required points is unknown in advance, using a Hammersley sequence is not recommended. Additionally, these sequences lose the iterative property of Halton sequences, which allows for the easy addition of points. Here is the figure representing the first 50, 250, and 500 points of a Hammersley sequence in base 2. Each subfigure illustrates the distribution of the points within the unit square $[0,1]^2$ (see figure 2.12)

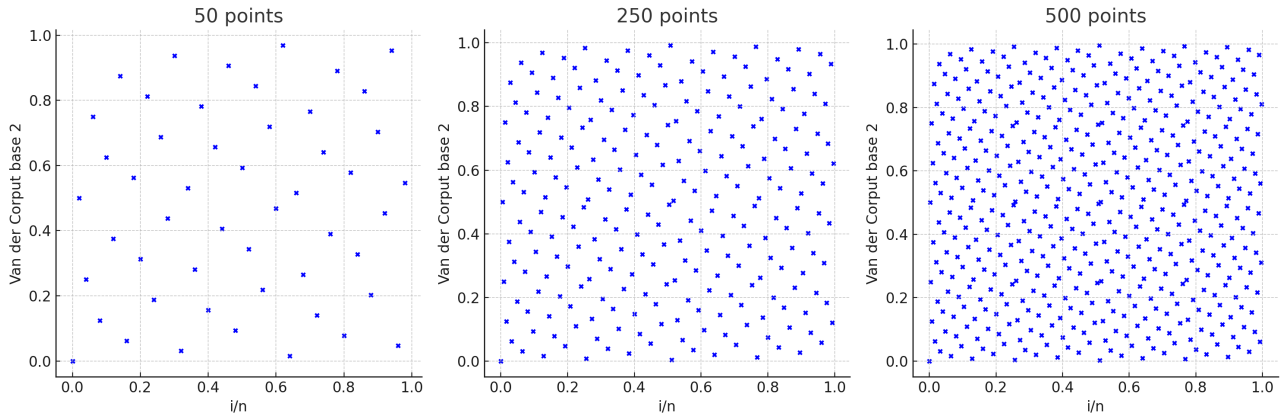

Figure 2.12: The first 50, 250, and 500 points of a Hammersley sequence in base 2.

2.2.4.3 Sobol' Sequences

Sobol sequences are defined based on primitive polynomials over the finite field $\mathbb{Z}_2 = \{0, 1\}$. Before proceeding, we recall the definition of a primitive polynomial.

Definition 2.12. A polynomial $p(t)$ of degree s of the form:

$$p(t) = t^s + u_{s-1}t^{s-1} + \cdots + u_1 t + u_0$$

is said to be **primitive** over the field $\mathbb{Z}_2$ if it meets the following conditions:

- It is **irreducible** over $\mathbb{Z}_2$, meaning it cannot be factored into lower-degree polynomials in $\mathbb{Z}_2[t]$.
- The smallest integer i such that $p(t)$ divides $t^i - 1$ (or $t^i + 1$) is exactly $2^s - 1$. This integer i is known as the **order** of the polynomial.

A primitive polynomial of degree s must include both the monomials t^s and 1, and it must contain an **odd** number of terms.

Definition 2.13. A **Sobol sequence** $S = \{x^0, x^1, x^2, \ldots, x^{n-1}\}$ in one dimension is defined as follows:

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

where $(p_1, p_2, \ldots, p_m)$ represents the binary expansion of i, and m is given by:

$$m = \begin{cases} 1 & \text{if } i = 0 \\ 1 + \lfloor \log_2 i \rfloor & \text{otherwise} \end{cases}$$

The symbol $\oplus$ denotes addition in $\mathbb{Z}_2$ [4].

For $k > s$, the coefficients l_k are computed using the recurrence relation:

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

where $u_1, u_2, \ldots, u_s$ are the coefficients of a primitive polynomial:$t^s + u_1 t^{s-1} + \cdots + u_{s-1} t + u_s$ defined over $\mathbb{Z}_2$. Additionally, the integers $l_1, \ldots, l_s$ must be **odd** and satisfy $1 \leq l_k \leq 2^k$ for $k = 1, \ldots, s$.

To generate a Sobol sequence in dimension d, it is sufficient to select d distinct primitive polynomials.

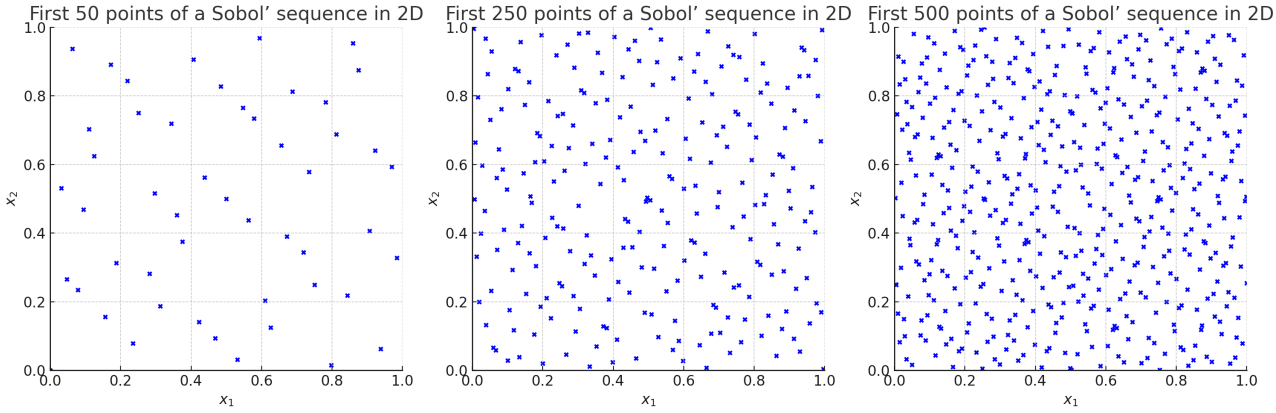

Figure 2.13: The first 50, 250, and 500 points of a Sobol' sequence in 2D

Discussion: Sobol sequences offer several significant advantages. Firstly, their construction is highly efficient since their binary nature aligns well with computer architectures, thereby reducing computation time [34]. Moreover, they generally maintain a well-balanced point distribution even as the dimensionality increases, unlike other low-discrepancy sequences that may suffer from degeneracy issues in high-dimensional spaces (Joe & Kuo, 2003 [35]).

2.2.4.4 Faure Sequences

Faure sequences are defined using the inverse radical function ϕ_b and a Pascal generator matrix C_{kl}, given by:

$$C_{k,l} = \begin{cases} \frac{(l-1)!}{(l-k)!(k-1)!}, & \text{if } k \leq l, \\ 0, & \text{otherwise.} \end{cases}$$

Definition 2.14. Let $b \geq d$ be a prime number. The Faure sequence $F = \{x_0, x_1, ..., x_{n-1}\}$ in dimension d is defined as:

[4] Addition modulo 2 is performed using an "exclusive or" (XOR).

$$x_i^{(j)} = \phi_b \left(\sum_{l=1}^{\infty} C_{j-1,l-1} i^l \mod b \right),$$

where $C_{j-1,l-1}$ represents the generator matrix of the j-th dimension of the Faure sequence in dimension d.

Remark 9. To achieve better uniform distribution, it is recommended to choose b as the smallest prime number greater than or equal to d.

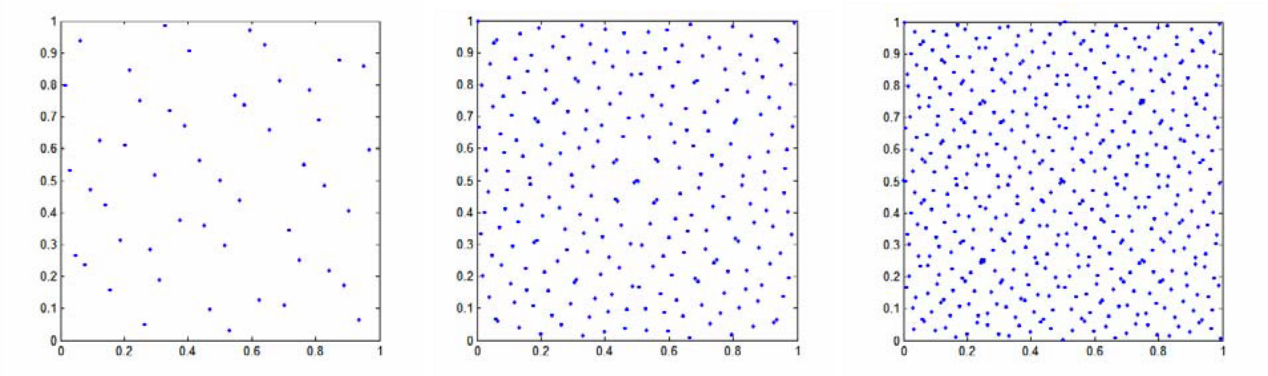

Figure 2.14: The first 50, 250, and 500 points of a Faure sequence in dimension 2.

Faure sequences are designed to ensure a locally uniform distribution of points.

2.3 Computer Experiments Designs

In this section, we introduce numerical designs. These designs are generated using the Markov Chain Monte Carlo method by utilizing point process stochastic processes.

2.3.1 Experimental Designs Based on the Strauss Process

One of the first stochastic models used for generating experimental designs is the Strauss point process, introduced by Franco et al.[36]. This process is defined by a conditional probability density given by:

$$\pi(x) = k\gamma^{s(x)}$$

where:

- k is a normalization constant,
- γ is an interaction parameter such that $0 < \gamma \leq 1$,

- $s(x)$ is the number of point pairs whose distance is below a threshold R:

$$s(x) = \sum_{i<j} \mathbf{1}_{\{\|x_i - x_j\| \leq R\}}$$

The parameter γ controls the repulsion between points:

If $\gamma = 1$, the process corresponds to a homogeneous Poisson process. If $\gamma < 1$, the points are more spaced out, introducing a repulsion effect.

2.3.2 Experimental Designs Based on the Marked Point Process

- **Marked point processes** [37]: extend the Strauss process by assigning each point x_i a **mark** m_i, which can represent additional information (prediction variance, factor importance, etc.). The probability density is given by:

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

where $\beta > 0$ is an intensity parameter and $n(x)$ is the number of points in the configuration.

The choice of marks is made by optimizing a predictive variance function, for example, for a polynomial model:

$$\text{var}(\hat{y}_{x_i}) = f(x_i)^T (F^T F)^{-1} f(x_i)$$

where F is the design matrix and $f(x_i)$ the regression vector.

- **Two-Mark Experimental Designs** [38]: In the specific case of **computer experimental designs with two marks**, we distinguish two types of points M_1 and M_2, each with its own interactions:

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

with:

  - $m_1(x)$: number of points of type M_1,
  - $m_2(x)$: number of points of type M_2,
  - $m_{11}(x)$: number of M_1-M_1 point pairs,

– $m_{12}(x)$: number of mixed M_1-M_2 pairs,

– $m_{22}(x)$: number of M_2-M_2 point pairs,

– $\gamma_{11}, \gamma_{12}, \gamma_{22}$: interaction coefficients.

2.3.3 Experimental Designs Based on Cluster Processes

Cluster random processes [39] introduce more complex neighborhood relationships between experimental points. An example is the continuous cluster random process, which defines the probability of a configuration x as:

$$\pi(x) = k\beta^{n(x)}\gamma^{-\mathrm{c}(x)}$$

where $\mathrm{c}(x)$ is the number of connected components in the graph defined by:

$$x_i \sim x_j \quad \text{if} \quad \|x_i - x_j\| \leq R$$

The Metropolis-Hastings algorithm is used here with a cluster movement dynamic, optimizing the coverage of the experimental space.

2.3.4 Experimental Designs Based on Area-Interaction Processes

The **area-interaction process** [40] is an interesting alternative where the interaction between points is defined based on the area covered by spheres of radius R centered on the experimental points. The process density is given by:

$$\pi(x) = k\beta^{n(x)}\gamma^{-m(U_R(x))}$$

where $m(U_R(x))$ is the Lebesgue measure of the union of spheres of radius R around the points, defined as:

$$U_r(x) = \bigcup_{i=1}^{n} B(x_i, r),$$

representing the density of the area occupied by the union of these balls. The process density is then expressed as:

$$\pi(x) = \alpha\, \beta^{n(x)}\, \gamma^{-m(U_r(x))},$$

with:

- $\alpha > 0$: normalization constant,
- $\beta > 0$: intensity parameter controlling the number of points,
- $\gamma > 0$: repulsion parameter penalizing the covered area,
- $m(U_r(x))$: Lebesgue measure (area) of $U_r(x)$.

  The measure $m(U_r(x))$ can be computed using the inclusion-exclusion formula:

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

CHAPTER 3

CRITERIA AND OPTIMAL DESIGNS

Studying the uniformity of point distributions is a challenging task, particularly in high-dimensional spaces where direct assessment often becomes impractical. As a result, it is essential to rely on specific criteria to determine whether a given distribution approximates uniformity and ensures good space-filling properties. These criteria are generally classified into three main categories.

First, *discrepancy criteria* measure the deviation between an empirical distribution and an ideal uniform distribution. They serve as a key indicator of point dispersion [33]. Discrepancy plays a central role in the theory of *quasi-Monte Carlo methods*, where low-discrepancy sequences such as those of Halton (1960)[27], Sobol' (1967)[29], and Faure (1982)[30] are widely used in numerical integration and computer experiment designs [41].

Second, *distance-based criteria* assess the regularity of a point distribution by comparing it to a regular grid [42]. This method is particularly useful in space-filling designs, where maintaining a minimum distance between points helps enhance interpolation accuracy and response surface modeling [43].

Third, the *entropy criterion* quantifies the amount of information contained in a design. Unlike the first two, it is model-dependent, relying on statistical assumptions about the response function [44]. Entropy-based criteria are often employed in *Bayesian experimental design*, where maximizing entropy leads to optimal information gain and reduced uncertainty in predictions [45]. Entropy is also closely related to interpoint distances, thus promoting well-distributed designs [46].

Each of these approaches supports the construction of optimal designs tailored to different objectives. Low-discrepancy designs provide uniform space coverage and are widely used in

numerical simulations and optimization [47]. Distance-based designs reduce clustering and are ideal for computer experiments [48].

Finally, for the *standard designs*, the quality of experimentation can be assessed using the model matrix even before running the experiments. This matrix depends on both the assumed mathematical model and the experimental point locations. It directly affects the prediction error, which should be minimized to a level comparable to the measurement error. Depending on the objective, various optimality criteria can be adoptedeither to ensure good domain coverage or to achieve precise estimation of model coefficients.

3.1 Optimality Criteria for computer experiments designs

3.1.1 Uniformity Criteria Based on Discrepancy and Low-Discrepancy Designs

3.1.1.1 Discrepancy

The fundamental definition of discrepancy was introduced in Section (2.2.4). As a reminder, discrepancy measures the deviation between a given point distribution and a uniform distribution; in other words, it quantifies the irregularity of the distribution. Below, we present the formal definition to clarify the underlying principle.

Remark 10. If the domain is reparameterizedfor instance, if we analyze $X_1^{(2)}$ instead of X_1then the objective will not be exactly the same. Ensuring uniformity for X_1 does not necessarily imply uniformity for $X_1^{(2)}$. This issue also arises in Bayesian inference when defining an informative prior distribution, where uniformity is not necessarily the primary goal.

Definition 3.1. (Niederreiter, 1987): Let X be a sequence of n points $x_1, \ldots, x_n$ in $[0,1]^d$, and let J be a subset of $[0,1]^d$. Using the previous notations, the discrepancy function is defined as:

$$D_n(J, X) = \frac{A(J, X)}{n} - \lambda^d(J)$$

where: - $A(J, X)$ is the number of indices i, $1 \leq i \leq n$, such that $x_i \in J$, - $\lambda^d(J)$ represents the Lebesgue measure (or volume) of J.

The extreme discrepancy of X, denoted as $D_n(X)$, is defined as:

$$D_n(X) = \sup_{J \in \mathcal{J}} D_n(J, X)$$

where $\mathcal{J}$ is the set of all subsets of $[0,1]^d$ of the form:$J = \prod_{i=1}^{d} [a_i, b_i]$. The star discrepancy of X, denoted as $D_n^*(X)$, is given by:$D_n^*(X) = \sup_{J \in \mathcal{J}^*} |D_n(J, X)|$

where $\mathcal{J}^*$ is the set of subsets of $[0,1]^d$ of the form:$\prod_{i=1}^{d} [0, b_i]$.

Remark 11. Let $\mu_X = \frac{1}{n} \sum_{i=1}^{n} \delta_{x_i}$ be the uniform probability measure on X. Then, $D_n(J, X)$ can be expressed as the distance between this measure and the Lebesgue measure, i.e.,

$$D_n(J, X) = |\mu_X(J) - \lambda_d(J)|$$

where $\lambda_d(J)$ denotes the Lebesgue measure (or volume) of J.

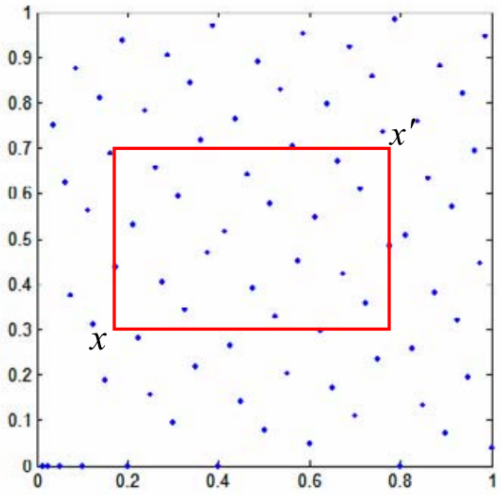

Figure 3.1: The first 80 points of a Hammersley sequencein demension 2 with a subset J defined by x and x' for extreme discrepancy

Consider the rectangle J defined by the corners: - $x = (0.2, 0.3)$ - $x' = (0.8, 0.7)$

The volume of this rectangle, given by the Lebesgue measure, is: $\lambda_d(J) = (0.8 - 0.2) \times (0.7 - 0.3) = 0.24$. This means that, under a perfectly uniform distribution, 24% of the points should ideally fall within J.

Now, considering a set X of 80 points, suppose that 18 points actually lie inside J. The observed proportion is:

$$\frac{A(J, X)}{n} = \frac{18}{80} = 0.225.$$

The discrepancy for this subset J is then given by:

$$D_n(J, X) = \left| \frac{A(J, X)}{n} - \lambda_d(J) \right| = |0.225 - 0.24| = 0.015.$$

By repeating this calculation for multiple subsets J and selecting the maximum value obtained, we derive the extreme discrepancy of the sequence X, denoted as $D_n(X)$. Similarly, $D_n^*(X)$ represents the star discrepancy. These discrepancies are defined in the L_∞-norm, which measures the worst-case deviation between the point distribution and the ideal uniformity.

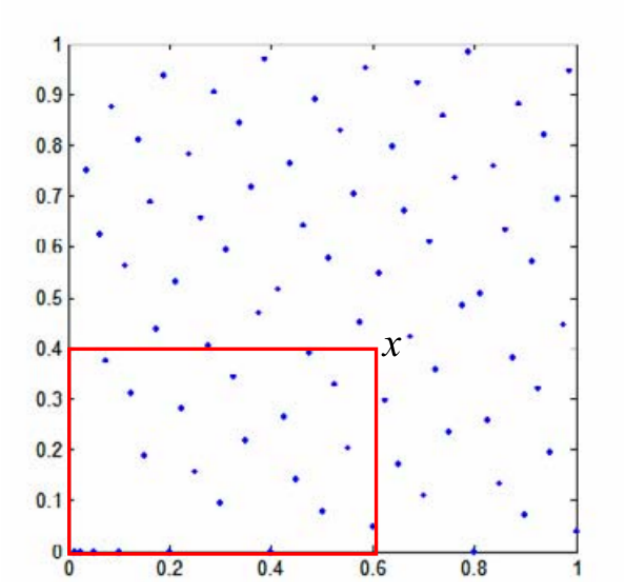

Figure 3.2: A subset J for the computation of the discrepancy at the origin

An alternative approach is to consider L_2-norm discrepancies, which provide a global measure of non-uniformity by integrating quadratic deviations over the entire domain. This concept will be further explored in Section (3.1.2).

Now that we have established a rigorous definition of discrepancy, we can explain why a regular grid can lead to poor discrepancy results. To achieve low discrepancy, it is crucial that the sampling uniformly covers all axis-aligned rectangles. However, a regular grid does not always meet this criterion. Some sub-rectangles may be poorly sampled due to the rigid structure of the grid. For instance, the placement of the boundaries relative to the grid may lead to significant deviations from ideal uniformity.

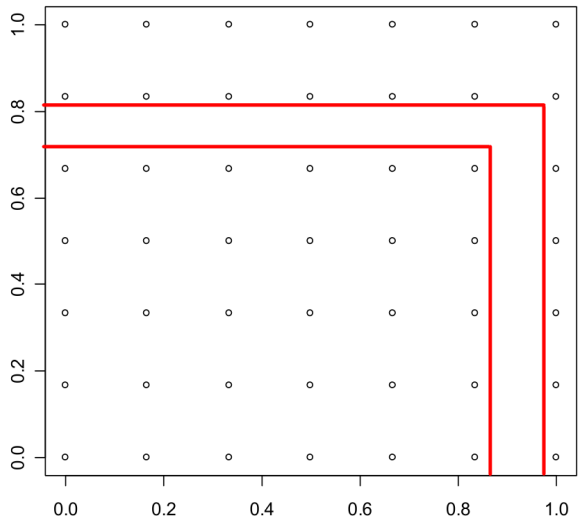

Figure 3.3: Factorial design and visualization of two axis-parallel rectangles not being uniformly sampled.

3.1.2 Discrepancy in L2-norm

The L_2-norm discrepancy is the only one that remains easily computable regardless of the dimension.

Let $X = \{x_1, \ldots, x_n\}$ be a sequence of n points in the interval $[0,1]^d$. This section first introduces the definition of various forms of L_2-norm discrepancy and then presents the corresponding computational approaches.

3.1.2.1 Discrepancy at Extremes and at the Origin

Definition 3.2. The L_2 discrepancy of a sequence of n points $x_1, \ldots, x_n$ in $[0,1]^d$ is defined as:

$$D^2_{L_2}(X_n) = \int_{[0,1]^d} D(J, X_n)^2 \, da \, db$$

where J represents subsets of $[0,1]^d$ of the form:

$$J = \prod_{i=1}^{d} [a_i, b_i]$$

Definition 3.3. The L_2 discrepancy at the origin of a sequence of n points $x_1, \ldots, x_n$ in $[0,1]^d$ is given by:

$$D^{*2}_{L_2}(X_n) = \int_{[0,1]^d} D(J, X_n)^2 \, db$$

where J represents subsets of $[0,1]^d$ of the form:

$$J = \prod_{i=1}^{d} [0, b_i]$$

3.1.2.2 Extreme and Origin Discrepancy Calculation

In dimension d, the values of $D^2_{L_2}(X_n)$ and $D^{*2}_{L_2}(X_n)$ can be computed using the following explicit formulas:

$$D^2_{L_2}(X_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{k=1}^{n} \prod_{j=1}^{d} \left(1 + \frac{1}{2} \max(x_{ij}, x_{kj}) - \frac{1}{2} \min(x_{ij}, x_{kj}) - x_{ij} x_{kj}\right) - \frac{1}{n} \sum_{j=1}^{d} \prod_{i=1}^{n} \left(\frac{1}{2} - x_{ij}\right) + \frac{1}{12^d}$$

and

$$D^{*2}_{L_2}(X_n) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{k=1}^{n} \prod_{j=1}^{d} (1 - \max(x_{ij}, x_{kj}) + x_{ij} x_{kj}) - \frac{1}{n} \sum_{j=1}^{d} \prod_{i=1}^{n} \left(\frac{1}{3} - x_{ij}\right) + \frac{1}{3^d}$$

3.1.2.3 Modified Discrepancy

Definition 3.4. The modified L_2-discrepancy of a sequence of n points $x_1, ..., x_n$ in $[0, 1]^d$ is defined as:

$$DL2^M_n(X_n) = \sum_{u \neq 0} \int_{[0,1]^u} D_p(J_u, X_n)^2 \, db_u$$

where $[0, 1]^u$ is the projection of the unit hypercube onto the components u, which form a subset of $\{1, ..., d\}$, with $p = \text{Card}(u)$. J_u denotes the projection of the subset J, defined as: $J_u = \prod_{i=1}^{d} [0, b_i]$

The modified L_2-discrepancy considers projections onto all subspaces and is defined by the following explicit formula:

$$DL2^M_n(X_n) = \left(\frac{4}{3}\right) - \frac{2^{1-d}}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \left(3 - (x^i_j)^2\right) + \frac{1}{n^2} \sum_{i=1}^{n} \sum_{k=1}^{n} \prod_{j=1}^{d} \left(2 - max\left(x^i_j, x^k_j\right)\right)$$

3.1.3 Centered Discrepancy

Definition 3.5. The centered L_2-discrepancy of a sequence of n points $x_1, \ldots, x_n$ in $[0, 1]^d$ is defined as:

$$D_{L_2}^C(X_n) = \sum_{u \neq 0} p_u \int_{[0,1]^u} D(J_u, X_n)^2 \, dx$$

where:

- $[0,1]^u$ represents the projection of the unit hypercube onto the components u, which is a subset of $\{1, ..., d\}$,
- $p = \text{Card}(u)$,
- J_u is the projection of a subset constructed from the considered point x and its nearest vertex.

Remark 12. In dimension 2, the set J_u can take four different forms, one of which is depicted in Figure 3.4. More generally, in dimension d, there are 2^d possible cases.

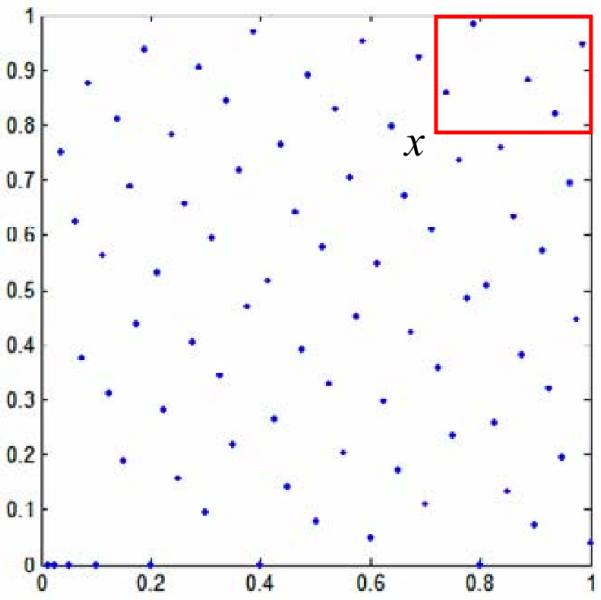

Figure 3.4: A subset J for the calculation of the centred discrepancy

Hickernell (1998) [42] provides an analytical expression for the centered discrepancy.

$$DL2_n^C(X)^2 = \left(\frac{13}{12}\right)^2 - \frac{2}{n}\sum_{i=1}^{n}\prod_{j=1}^{d}\left(1 + \frac{1}{2}|x_j^i - 0.5| - \frac{1}{2}|x_j^i - 0.5|^2\right)$$
$$+\frac{1}{n^2}\sum_{i=1}^{n}\sum_{k=1}^{n}\prod_{j=1}^{d}\left(1 + \frac{1}{2}|x_j^i - 0.5| + \frac{1}{2}|x_j^k - 0.5| - \frac{1}{2}|x_j^i - x_j^k|\right)$$

3.1.4 Symmetric Discrepancy

Definition 3.6. The L_2 symmetric discrepancy of a sequence of n points $x_1, ..., x_n$ in $[0,1]^d$ is defined as:

$$DL_2^S(X) = \sum_{u \neq 0} \int_{[0,1]} D(J_u, X)^2 \, dx$$

where J_u is the projection of the interval J onto the subspace defined by the components u, and J represents the union of symmetric subsets, i.e., subsets where the sum of the coordinates of the vertices remains equal.

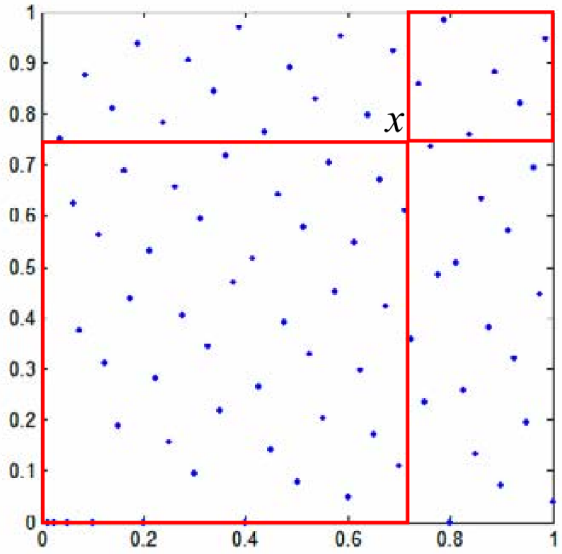

Figure 3.5: For x = (0.7, 0.75), the total volume of the two subsets J is 0.6, and the total proportion of points is 49/80 = 0.6125. The difference between these two values is therefore 0.0125

We also have an analytical formula to compute this discrepancy:

$$DL2_n^S(X)^2 = \left(\frac{4}{3}\right)^d - \frac{2}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \left(1 + 2x_j^i - 2(x_j^i)^2\right) + \frac{2^d}{n^2} \sum_{i=1}^{n} \sum_{k=1}^{n} \prod_{j=1}^{d} \left(1 - |x_j^i, x_j^k|\right)$$

3.1.5 Low-Discrepancy Designs

The concept of low-discrepancy designs discussed here differs from that of sequences detailed in Section 2.2.4 for the following reasons.

First, let us clearly define what we mean by low-discrepancy designs. These designs refer to stochastic configurations generated by a simple exchange algorithm, which favors arrangements

that minimize discrepancy.

Numerous results exist for bounding discrepancy. In Section 2.2.4, we mentioned the Koksma-Hlawka inequality, but there are also specific upper bounds for each of the previously discussed sequences. For instance, Faure[32] demonstrated that the star discrepancy of a n-point Halton sequence in d dimensions, generated using bases $b_1, ..., b_d$, is bounded by:

$$\frac{d}{n} + \frac{1}{n} \prod_{j=1}^{d} \frac{b_j - 1}{2 \log b_j} \log n + \frac{b_j + 1}{2}.$$

Faure's theorem [49] further suggests that discrepancy can be reduced by generalizing sequences, specifically by applying permutations to sequence elements. Consequently, the idea of obtaining low-discrepancy designs through a simple exchange algorithm appears not only feasible but also promising.

Theorem 3.1.1. *Let X be the generalized Van der Corput sequence in base $b = 12$, generated by the permutation*

$$\sigma = (0, 5, 9, 3, 7, 110, 4, 8, 2, 6, 11)$$

Then, we obtain:

$$\limsup_{n \to \infty} \frac{n D_n^*(X)}{\log n} \approx 0.224.$$

The study of low-discrepancy designs is of dual interest. Firstly, it demonstrates that it is possible to obtain designs with a lower discrepancy than most of the sequences discussed in Section 2.2.4. Secondly, these inherently non-deterministic designs avoid the projection defects commonly found in high-dimensional sequences due to their regular structure.

The construction of these designs relies on discrepancy computation, and they are primarily built using the L_2-norm discrepancy, which is significantly easier to compute than the L_∞-norm discrepancy. This approach makes it possible to design plans based on any of the discrepancies defined in Section 3.1.2, particularly the centered or modified discrepancy, which consider point projections on the margins. Consequently, in the following discussion, the designs will be generated using the centered discrepancy.

However, a major drawback of these designs is their computational cost. Even though the L_2-norm discrepancy can be calculated using simple analytical formulas, low-discrepancy sequences are much faster to generate. For instance, a Halton sequence with 600 points in 60 dimensions can be produced instantly, whereas constructing a low-discrepancy design of similar

size is computationally prohibitive.

3.1.6 Distance Criteria and Optimal Designs

Distance-based criteria aim to assess the proximity between a given point distribution and that of a regular grid. In this section, we will focus on the most commonly used uniformity criteria, which are based on the distance between neighboring points. For a more in-depth discussion of additional uniformity measures, the reader may refer to Gunzburger [50].

The idea is to generate designs whose points are close to a regular grid without exactly matching it, in order to avoid undesirable misalignments. Our goal is to construct designs with a quasi-periodic distribution, striking a balance between a regular grid and good uniformity, often measured using discrepancy criteria.

The distance between two points x_i and x_k, denoted as $\text{dist}(x_i, x_k)$, is given by the Euclidean distance:

$$\text{dist}(x_i, x_k) = \left[\sum_{j=1}^{d} (x_j^i - x_j^k)^2 \right]^{1/2}$$

3.1.6.1 Covering Measure

Definition 3.7. Let $X = \{x_1, ..., x_n\} \subset [0,1]^d$ be a sequence of n points in a d-dimensional space.

The covering measure λ is defined as:

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

where: - $\gamma_i = \min_{k \neq i} \text{dist}(x^i, x^k)$ represents the minimum distance between point x_i and the other points in the sequence. - $\bar{\gamma} = \frac{1}{n} \sum_{i=1}^{n} \gamma_i$ is the average of all γ_i.

Interpretation

If the points are arranged on a regular grid, then $\gamma_i = \gamma$ for all i, leading to $\lambda = 0$.

Therefore, the smaller λ, the closer the points are to a regular grid. This expression explicitly highlights the coefficient of variation of the sample γ_i, which is the ratio of the standard deviation to the mean.

3.1.6.2 The Distance Ratio

Definition 3.8. Let $X = \{x_1, ..., x_n\} \subset [0,1]^d$ be a set of n points in d-dimensional space.

The *distance ratio* is defined as:

$$R = \frac{\max\limits_{i=1,...,n} \gamma_i}{\min\limits_{i=1,...,n} \gamma_i}$$

where

$$\gamma_i = \min_{k \neq i} \text{dist}(x^i, x^k)$$

represents the minimum distance between the point x_i and any other point in the set.

When the points are arranged on a regular grid, we have $\gamma_i = \gamma$ for all i, leading to

$$R = \frac{\max \gamma_i}{\min \gamma_i} = 1.$$

Therefore, the closer R is to 1, the more the point distribution resembles a regular grid.

3.1.6.3 Maximin and Minimax Distances

Johnson et al.[48] introduced the *maximin* and *minimax* distances to construct designs that optimize space-filling properties.

Definition 3.9. [51]

These criteria are defined using the Euclidean distance:

- Maximin Distance (MinDist):

$$\text{MinDist} = \min_{x_i \in X} \min_{\substack{x_k \in X \\ k \neq i}} \text{dist}(x^i, x^k)$$

- Minimax Average Distance (AvgDist):

$$\text{AvgDist} = \frac{1}{n} \sum_{i=1}^{n} \min_{\substack{x_k \in X \\ k \neq i}} \text{dist}(x^i, x^k)$$

where $X = \{x_1, ..., x_n\}$ represents an experimental design with n points in d-dimensional space.

3.1.7 Entropy Criterion and Maximum Entropy Designs

This criterion differs from the previously presented ones as it does not directly assess the uniformity or space-filling properties of a design in an exploratory phase. Indeed, entropy calculation is generally feasible only when the underlying distribution is known, an assumption that is often not met in exploratory settings.

The purpose of introducing entropy here is to lay the foundation for a method of optimal design generation based on this criterion (see section 3.1.7.2). Although this criterion is not inherently linked to spatial uniformity, the resulting designs exhibit good space-filling properties. Additionally, it allows for the consideration of variable anisotropy, which can sometimes be inferred during the exploratory phase based on prior knowledge of the physical phenomenon.

3.1.7.1 Definition of Entropy

Shewry and Wynn (1987)[44] described entropy as "the amount of information contained in an experiment." More generally, entropy quantifies the information content within a probability distribution.

Definition 3.10. The entropy of a continuous random variable X with probability density function f is given by:

$$H(X) = -\int_{x\in\mathbb{R}} f(x)\log f(x)\,dx = -E_X(\log f(X))$$

with the convention $0\ln(0) = 0$.

For mathematical simplicity, we use the natural logarithm. This choice does not affect the results, as entropy is merely translated by a constant factor.

Similarly, for a continuous random vector $X = (X_1, ..., X_d)$ in $\mathbb{R}^d$ with density f, entropy is defined as:

$$H(X) = -\int_{\mathbb{R}^d} f(x)\log f(x)\,d\mu(x),$$

where μ is the Lebesgue measure.

Remark 13. Entropy depends solely on the probability density function f and not on the specific values taken by X. Consequently, it cannot be directly computed from an experimental design.

Maximizing Entropy for Experimental Design

The goal is to select an experiment e from a set E that maximizes the expected information gain. Shewry and Wynn (1987) [44] highlighted several challenges with this approach, partic-

ularly regarding the definition of E. They proposed considering E as a finite set of possible experiments and established a connection between information gain and entropy.

If the experimental domain E consists of N points, each associated with a response Y_i for $i = 1, ..., N$, we can partition E into two subsets: - D, the chosen design points, - D^c, the remaining points.

The standard decomposition of entropy yields:

$$H(Y_E) = H(Y_D) + E_{Y_D}[H(Y_{\bar{D}}|Y_D)].$$

The term $\mathbb{E}[H(Y_{D^c}|Y_D)]$ corresponds to the expected reduction in entropy when selecting D. Maximizing entropy-based designs thus involves choosing D to maximize $H(Y_D)$, the entropy of the selected design points.

The purpose of maximum entropy designs is therefore to maximize the information gained from experiments relative to a parameter θ. Many studies have explored this concept, notably those by Koehler and Owen [52] and Santner et al. [9]. Entropy-based experimental designs have been widely used to approximate complex deterministic models, as discussed by Mitchell and Scott [53], Currin et al. [54], and Sebastiani and Wynn [55].

Remark 14. This approach combines prior knowledge with experimental data under an assumed model to produce a posterior distribution, placing it entirely within the Bayesian framework. For a comprehensive review of Bayesian experimental designs, see Chaloner and Verdinelli [45].

3.1.7.2 Maximum Entropy Designs

The general definition of maximum entropy designs typically requires knowing the response values at the design points, which means that entropy is not, in principle, an intrinsic criterion. However, Shewry and Wynn [44] proposed a formulation that allows constructing such designs without needing the actual response values.

The method they introduced focuses on space-filling by distributing points according to a spatial correlation matrix. In the specific case of a centered Gaussian process f, Shewry and Wynn (1987) showed [1] that the entropy $H(Y(X))$ depends directly on the determinant of the covariance matrix:

$$H(Y(X)) \propto \ln \det(C(X))$$

[1] This demonstration is excellently detailed in Koelher and Owen [52]

where $C(X)$ is the covariance matrix. Therefore, under the assumption of stationarity, generating a maximum entropy design amounts to maximizing the determinant of the correlation matrix.

Remark 15. If the model is linear, then the determinant can be expressed in terms of the design matrix. In this case, a design obtained using a classical approach, such as an exchange algorithm, would be D-optimal.

This equivalence holds only if the responses at the design points follow a multivariate normal distribution, without any specific assumptions on the covariance structure.

Let

$$X = (X_1, \ldots, X_n)^T$$

be a vector of random variables. The variancecovariance matrix of X is given by:

$$C(X) = \begin{pmatrix} \sigma_1^2 & \operatorname{cov}(X_1, X_2) & \cdots & \operatorname{cov}(X_1, X_n) \\ \operatorname{cov}(X_2, X_1) & \sigma_2^2 & \cdots & \operatorname{cov}(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ \operatorname{cov}(X_n, X_1) & \operatorname{cov}(X_n, X_2) & \cdots & \sigma_n^2 \end{pmatrix}$$

where σ_i is the standard deviation of X_i, and

$$\operatorname{cov}(X_i, X_j) = \sigma_i \sigma_j \rho_{ij}$$

is the covariance between X_i and X_j.

If the variables X_i are standardized, the covariance matrix $C(X)$ becomes the correlation matrix:

$$C(X) = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{pmatrix}$$

Now, let us define a spatial correlation matrix $C = [\rho_{ij}]$ as follows:

$$\rho_{ij} = \begin{cases} 1, & \text{if } i = j \\ 1 - \gamma(h_{ij}), & \text{if } h_{ij} < a \\ 0, & \text{if } h_{ij} > a \end{cases}$$

Here, $\gamma(h)$ is the variogram, h_{ij} denotes the Euclidean distance between points i and j, and a is the range parameter of the variogram

We can then compute, for

$$X = (x_1, \ldots, x_n)$$

a vector of points in a d-dimensional space, a spatial correlation matrix defined as:

$$C(X) = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1n} \\ \rho_{21} & 1 & \cdots & \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{n1} & \rho_{n2} & \cdots & 1 \end{pmatrix}$$

where ρ_{ij} is a function of the distance between points i and j, computed based on a spatial correlation model that is assigned a priori to the experimental space (see equation (1) above).

The determinant of $C(X)$ reaches its maximum when $\rho_{ij} = 0$, that is, when each pair of points is separated by a distance greater than the range a of the spatial correlation function.

Thus, the goal is to maximize the determinant of $C(X)$ using an exchange algorithm, such as those proposed by Fedorov or Mitchell, as described below:

Algorithm 1: Procedure for Generating a Maximum Entropy Design (DETMAX)

Input: Number of points n, maximum number of iterations $N_{\max}$, variogram model

Output: A design X with (approximately) maximum entropy

Initialize: Randomly select an initial design $X^{(0)}$ of n points in $[0,1]^d$ and fix a variogram model;

Compute $\det(C(X^{(0)}))$;

for $k = 1$ **to** N_{max} **do**

  Randomly choose an index $i \in \{1, \ldots, n\}$;

  Simulate a new point z_i uniformly in $[0,1]^d$;

  Let $X^{(k)}$ be $X^{(k-1)}$ with x_i replaced by z_i;

  if $\det(C(X^{(k)})) > \det(C(X^{(k-1)}))$ **then**

    Accept the new design $X^{(k)}$;

  end

  else

    Reject the update and keep $X^{(k)} = X^{(k-1)}$;

  end

end

return $X^{(N_{max})}$

Remark 16. The classical DETMAX algorithm explores all points on a regular grid.

3.2 Optimality Criteria for Standard Designs

A good experimental design is one that minimizes the prediction error on the responses. A general rule is that the prediction error should be of the same order of magnitude as the measurement error on the observed responses. Depending on the selected optimality criterion, the location of the experimental points may vary from one design to another.

Several optimality criteria exist. Some focus on the distribution of the variance across the experimental domainsuch as the rotational isovariance criterion. Others aim to ensure that the resulting mathematical model is of high quality. These criteria are primarily concerned with the precision of the models estimated coefficients.

3.2.1 A-Optimality Criterion

An experimental design matrix is said to be **A-optimal** if it minimizes the trace of the dispersion matrix:

$$\mathrm{Tr}\left((X^T X)^{-1}\right)$$

This criterion focuses on minimizing the average variance of the estimated coefficients.

3.2.2 D-Optimality Criterion

A design matrix is **D-optimal** if it minimizes the determinant of its dispersion matrix:

$$\det\left((X^T X)^{-1}\right)$$

Equivalently, this maximizes the determinant of the information matrix $X^T X$, reducing the volume of the confidence ellipsoid for the model coefficients.

3.2.3 E-Optimality Criterion

A design matrix is **E-optimal** if it minimizes the largest eigenvalue of the dispersion matrix $(X^T X)^{-1}$. This criterion ensures that the worst-case variance among the coefficients is as small as possible.

3.2.4 G-Optimality Criterion

The **G-optimality** criterion considers the maximum prediction variance across the design domain:

$$d = \max_{u \in \mathcal{D}} \hat{y}^2(u)$$

The best design under this criterion minimizes d.

G-Efficiency The **G-efficiency** of a design is calculated as:

$$\text{Eff}_G = \frac{100 \cdot q}{N \cdot d_{\max}}$$

where q is the number of model parameters, N is the number of experiments, and $d_{\max}$ is the maximum prediction variance. A design is close to G-optimal if $\text{Eff}_G \approx 100\%$.

3.2.5 M-Criterion

The **M-criterion** assesses the information quality of a design, independent of the number of runs. The moment matrix is defined as:

$$M = \frac{1}{N} X^T X$$

Let M_1 and M_2 be the moment matrices for two designs with N_1 and N_2 runs:

$$M_1 = \frac{1}{N_1} X_1^T X_1, \quad M_2 = \frac{1}{N_2} X_2^T X_2$$

Design 1 is more efficient than Design 2 with respect to this criterion if $det(M_1) > det(M_2)$.

3.2.6 Orthogonality Criterion

A design is **orthogonal** if it leads to independent coefficient estimations, which occurs when the information matrix $X^T X$ (or its inverse) is diagonal. This implies zero covariance between coefficients.

3.2.7 Near-Orthogonality Criterion

The **near-orthogonality** criterion is satisfied when the submatrix (excluding the first row and column) of $(X^T X)^{-1}$ is diagonal, indicating that all coefficients except the intercept are nearly

uncorrelated.

3.2.8 Iso-Variance by Rotation Criterion

This criterion requires the prediction error to be constant for all points located at equal distances from the center of the experimental domain. It ensures isotropy and rotational symmetry of the variance distribution across the space.

CHAPTER 4

COMPARATIVE ANALYSIS OF STANDARD AND COMPUTER EXPERIMENTAL DESIGNS BASED ON OPTIMALITY CRITERIA

This chapter is dedicated to the comparative analysis between standard design of experiments (such as factorial designs, composite designs, Box-Behnken designs, and so on.) and computer designs generated through simulation (such as low-discrepancy sequences, maximin distance designs, Strauss designs, marked Strauss designs, and so on.). Based on the optimality criteria discrepancy, inter-point distances, and entropy we evaluate the relative performance of each type of design. This comparison aims to identify scenarios where one design may be more appropriate than another, depending on the experimental goals pursued, such as space-filling, estimation accuracy.

4.1 Comparison of Standard Experimental Designs According to Classical Optimality Criteria

This section focuses on the comparative analysis of several classical experimental designs based on widely accepted optimality criteria. The designs examined include full factorial designs at two and three levels, Mozzo designs, composite designs, and Box-Behnken designs. These designs are commonly used in practice due to their simplicity and effectiveness in modeling first- or second-order responses.

Each design is evaluated using key optimality criteria such as:

- **D-optimality**, based on the determinant of the information matrix ($X^T X$),

- **A-optimality**, based on the trace of $(X^T X)^{-1}$,
- **G-optimality**, through the maximum prediction variance $\max d(x)$,
- **M-criterion**, related to the determinant of the moment matrix $M = \frac{1}{N} X^T X$
- **G-efficiency**, indicating the overall predictive quality of the design.

Through this analysis, we aim to highlight the trade-offs between the number of experiments, estimation precision, and spatial coverage of the experimental domain. The comparison allows us to determine under which conditions each design performs best and to provide guidance for choosing the most suitable design according to the study objectives.

4.1.1 Comparison for Two-Factor Designs

We examine four standard experimental designs used in the case of two factors: the full factorial designs at two and three levels, the Mozzo design, and the composite design. These designs are evaluated according to several classical optimality criteria such as D-optimality (through $\det(X^T X)$), A-optimality (trace of the inverse information matrix), G-optimality (maximum prediction variance), and G-efficiency. The goal is to assess the trade-offs between model estimation quality, prediction accuracy, and space-filling properties.

Table 4.1: Comparison of standard designs for two factors

Design	**Full Factorial 2^k**	**Full Factorial 3^k**	**Mozzo**	**Composite**
Nb. Runs.	6	9	3	12
Nb. Levels	2 ; 2	3 ; 3	3 ; 3	5 ; 5
$\det(X^T X)^{-1}$	2.61×10^{-3}	1.92×10^{-4}	0.129	3.05×10^{-5}
$\text{trace}[(X^T X)^{-1}]$	0.92	2.1389	1.5771	2.18
M	0.296	9.754×10^{-3}	0.287	6.16×10^{-2}
$\max d(x)$	0.805	0.81	1.564	0.99
G-efficiency (%)	82.81	82.30	63.39	67.34

The analysis of the results in Table 4.1 highlights several trade-offs. The full factorial design at two levels shows the highest D-optimality and a very good G-efficiency, indicating strong model identifiability with a relatively small number of experiments. The factorial 3^k design offers a finer resolution due to more levels but slightly reduces G-efficiency and increases the complexity of the experiment.

Although the Mozzo design is highly economical (with only 3 points), it performs poorly in terms of D- and A-optimality and the highest maximum prediction variance, which limits its

practical utility for accurate modeling. The composite plan, although it uses the highest number of experiments (12), performs moderately across most criteria, offering a balance between coverage and estimation quality, but with lower G-efficiency.

For two-factor models, the full factorial design at two levels appears to offer the best compromise between economy and statistical robustness.

4.1.2 Comparison for Three-Factor Designs

A comparative evaluation of three classical designs used in the case of three factors: the full factorial design at three levels, the central composite design, and the Box-Behnken design. These designs are commonly used when second-order models are required, particularly for response surface methodology. The comparison is based on optimality criteria such as D-optimality, A-optimality, moment matrix determinant, G-optimality, and G-efficiency.

Table 4.2: Comparison of standard designs for three factors

Design	**Full Factorial (3 levels)**	**Composite**	**Box-Behnken**
Nb. Runs.	27	15	15
Nb. Levels	3 ; 3 ; 3	5 ; 5 ; 5	3 ; 3 ; 3
$\det(X^T X)^{-1}$	1.70×10^{-11}	1.19×10^{-9}	3.97×10^{-8}
$\text{trace}[(X^T X)^{-1}]$	1.176	1.769	2.270
M	2.85×10^{-4}	1.452×10^{-2}	4.364×10^{-5}
$\max d(x)$	0.51	0.525	0.73
G-efficiency (%)	72.62	67.34	59.52

The results in Table 4.2 show that the full factorial design at three levels achieves the best overall performance in terms of both D-optimality and A-optimality. This confirms its strong ability to estimate complex models with high precision, at the cost of a higher number of experimental runs (27).

The composite design, with only 15 runs, achieves a good compromise. Although its D-efficiency is slightly lower, it still provides reasonable estimation accuracy with improved economy. Moreover, its G-efficiency remains acceptable, reflecting reliable prediction performance over the experimental space.

The Box-Behnken design, also with 15 runs, shows lower efficiency on all statistical criteria. While it is often appreciated for requiring fewer experimental runs and for being rotatable, it presents the largest prediction variance and the lowest G-efficiency among the three, which could be a limiting factor in some applications.

For three-factor experiments aiming at quadratic modeling, the full factorial design provides the best estimation power, while the composite plan remains a viable alternative when experimental cost must be minimized.

4.1.3 Comparison for Four-Factor Designs

We compare several standard experimental designs suitable for four factors. These include full factorial designs at two and three levels, the Box-Behnken design, the composite design, as well as the Mozzo and Doehlert designs. The evaluation is carried out using multiple optimality criteria, including D- and A-optimality, maximum prediction variance, and G-efficiency. These criteria help assess the balance between estimation accuracy, prediction robustness, and space-filling properties.

Table 4.3: Comparison of standard designs for four factors (Part 1)

Design	**Full factorial 3^4**	**Full factorial 2^4**	**Box-Behnken**
Nb. Runs.	81	16	27
Nb. Levels	3 ; 3 ; 3 ; 3	2 ; 2 ; 2 ; 2	3 ; 3 ; 3 ; 3
$\det(X^TX)^{-1}$	6.52×10^{-11}	1.67×10^{-3}	1.96×10^{-6}
$\text{trace}[(X^TX)^{-1}]$	0.574	0.3125	2.9167
$\max d(x)$	22.50	5.00	15.75
G-efficiency (%)	66.67	100.00	95.24

Table 4.4: Comparison of standard designs for four factors (Part 2)

Design	**Composite Design**	**Mozzo Design**	**Doehlert Design**
Nb. Runs.	30	18	24
Nb. Levels	3 ; 3 ; 3 ; 3	5 ; 5 ; 5 ; 5	5 ; 5 ; 5 ; 5
$\det(X^TX)^{-1}$	1.96×10^{-9}	Not valid	2.24
$\text{trace}[(X^TX)^{-1}]$	0.8542	Not valid	64.45
$\max d(x)$	17.50	Not available	19.20
G-efficiency (%)	85.71	Not available	78.12

The comparison reveals distinct performance patterns among the designs. The full factorial 2^4 design achieves the best G-efficiency (100%) and lowest A-optimality trace, but it lacks the resolution to estimate second-order effects fully. The 3^4 factorial design provides better model estimation at the cost of a high number of runs (81), with good D- and A-optimality, although it is less space-filling.

The composite design represents a strong compromise, balancing estimation accuracy and point dispersion, with high G-efficiency and moderate prediction variance. Box-Behnken performs acceptably but shows lower estimation quality, as reflected in its A-optimality.

The Doehlert design, while efficient in terms of point distribution and fewer experimental runs, yields poor A-optimality and an unusually high prediction variance, limiting its suitability for accurate modeling. The Mozzo design values appear numerically unstable (invalid determinants or traces), suggesting it may not be adequate for four-factor second-order modeling.

The composite design offers the best compromise between estimation quality and space-filling performance in the case of four factors.

4.1.4 Comparison for Five-Factor Designs

We compare several standard experimental designs suitable for five-factor studies. The considered designs include full factorial designs at two and three levels, the Box-Behnken design, the central composite design, and the Doehlert design. These are evaluated according to optimality criteria such as D-optimality, A-optimality, G-efficiency, and the maximum prediction variance.

Table 4.5: Comparison of standard designs for five factors (Part 1)

Design	**Full factorial 3^5**	**Full factorial 2^5**	**Box-Behnken**
Nb. Runs.	243	32	43
Nb. Levels	3 ; 3 ; 3 ; 3 ; 3	2 ; 2 ; 2 ; 2 ; 2	3 ; 3 ; 3 ; 3 ; 3
$\det(X^TX)^{-1}$	2.14×10^{-19}	1.15×10^{-4}	6.96×10^{-9}
$\text{trace}[(X^TX)^{-1}]$	0.2613	0.1875	3.9271
$\max d(x)$	33.50	6.00	21.50
G-efficiency (%)	62.69	100.00	97.67

Table 4.6: Comparison of standard designs for five factors (Part 2)

Design	**Composite Design**	**Doehlert Design**
Nb. Runs.	48	34
Nb. Levels	3 ; 3 ; 3 ; 3 ; 3	5 ; 5 ; 5 ; 5 ; 5
$\det(X^TX)^{-1}$	1.18×10^{-15}	146.33
$\text{trace}[(X^TX)^{-1}]$	0.6887	162.94
$\max d(x)$	26.91	26.81
G-efficiency (%)	78.04	78.34

From the comparison above, it is clear that the full factorial design at two levels offers the best G-efficiency (100%) and the lowest prediction variance, although it cannot model second-order effects.. The 3^5 factorial design offers excellent D- and A-optimality, but at the cost of significant computational expense due to the very high number of experiments (243).

The composite design presents a strong compromise, maintaining relatively low prediction variance and acceptable G-efficiency while reducing the number of runs compared to the full

factorial. The Box-Behnken design, with 43 runs, also performs very well in terms of G-efficiency and space coverage, although with reduced estimation accuracy.

The Doehlert design is the most economical in terms of prediction dispersion, but it exhibits extremely poor A-optimality, making it less appropriate for accurate model estimation in higher-dimensional settings.

For five-factor designs, the composite and Box-Behnken plans appear to provide a practical balance between statistical quality and experimental cost.

4.1.5 Comparison for Six-Factor Designs

A detailed comparison of several standard experimental designs suitable for six factors. The criteria considered include the number of experiments, the number of levels, statistical precision indicators such as the determinant and the trace of the information matrix $(X^tX)^{-1}$, the maximum distance between points, as well as G-efficiency. These criteria allow for the evaluation of trade-offs between experimental cost, estimation accuracy, and the space-filling quality.

Table 4.7: Comparison of standard designs for six factors (part 1)

Design	**Full factorial 3^6**	**Full factorial 2^6**	**Box-Behnken**
Nb. Runs.	729	64	63
Nb. Levels	3	2	3
$\det(X^TX)^{-1}$	1.37×10^{-31}	1.4×10^{-6}	4.37×10^{-12}
$\text{trace}[(X^TX)^{-1}]$	0.113512	0.109375	5.220833
M	9.21×10^{-5}	1	1.83×10^{-11}
$\max d(x)$	46.75	7	28.35
G-efficiency (%)	59.89	100	98.77

Table 4.8: Comparison of standard designs for six factors (part 2)

Design	**Composite Design**	**Doehlert Design**
Nb. Runs.	82	46
Nb. Levels	3	5
$\det(X^TX)^{-1}$	6.53×10^{-24}	1.44×10^{5}
$\text{trace}[(X^TX)^{-1}]$	0.514174	351.4375
M	0.38	1
$\max d(x)$	44.8071	35.9375
G-efficiency (%)	62.49	77.91

Comparative Analysis The full factorial design 3^6 provides excellent coverage of the experimental space ($\max d(x) = 46.75$) and good statistical precision (trace = 0.113512), but at the cost of a very high experimental burden (729 runs), making it impractical in many cases.

The 2^6 factorial design offers a highly efficient solution: with only 64 runs, it achieves maximum G-efficiency (100%) and optimal precision (trace = 0.109375). However, its limited spatial dispersion ($\max d(x) = 7$) reduces its ability to explore the factor space thoroughly.

The Box-Behnken design (63 runs) represents a good compromise. It offers excellent G-efficiency (98.77%) and acceptable dispersion ($\max d(x) = 28.35$), though with lower statistical precision (higher trace value).

The central composite design (82 runs) provides good dispersion ($\max d(x) = 44.81$) and satisfactory precision (trace = 0.514174) at a moderate cost, but has relatively low G-efficiency (62.49%).

Finally, the Doehlert design performs poorly in terms of statistical precision (trace = 351.4375) and exhibits an abnormally high determinant, indicating multicollinearity issues. Despite its low experimental cost (46 runs), this design is not well suited for accurately modeling complex phenomena involving six factors. **Conclusion**

In summary, the 2^6 factorial and Box-Behnken designs offer the best trade-offs between cost, precision, and space-filling ability. The final choice depends on the required level of precision and specific experimental constraints.

4.1.6 Comparison for Seven-Factor Designs

This section presents a comparative evaluation of several classical experimental designs used to study seven factors. The evaluated criteria include the number of experiments, number of levels, determinant and trace of the information matrix, maximum pairwise distance between design points, and G-efficiency. The objective is to identify the most suitable designs under cost constraints and modeling requirements.

Table 4.9: Comparison of standard designs for seven factors

Design	**Full factorial 3^7**	**Full factorial 2^7**	**Box-Behnken**	**Composite**
Nb. Runs.	2187	128	87	148
Nb. Levels	3	2	3	3
$\det(X^TX)^{-1}$	3.14×10^{-47}	1.56×10^{-8}	6.81×10^{-15}	1.29×10^{-34}
$\text{trace}[(X^TX)^{-1}]$	0.047668	0.0625	6.779167	0.345315
M	5.24×10^{-6}	1	8.78×10^{-17}	2.82
$\max d(x)$	62.25	8	36.25	80.2488
G-efficiency (%)	57.83	100	99.31	44.86

Comparative analysis The full factorial design 3^7 offers excellent accuracy (trace = 0.047668) and very broad space coverage ($\max d(x) = 62.25$), but it is extremely expensive, requiring 2187

experiments, which is often impractical.

The 2^7 full factorial design is much more economical (128 experiments), achieves the highest G-efficiency (100%), and has good statistical precision (trace = 0.0625). However, its low spatial dispersion ($\max d(x) = 8$) limits global space exploration.

The Box-Behnken design, with only 87 experiments, provides very high G-efficiency (99.31%) but suffers from lower precision (trace = 6.779167). Still, it remains a good trade-off for moderately complex response surfaces.

The composite design (148 runs) improves space coverage ($\max d(x) = 80.25$), but its low G-efficiency (44.86%) and moderate precision (trace = 0.345315) reduce its appeal for applications requiring high statistical quality.

For seven-factor experiments, the 2^7 factorial design stands out as an excellent option due to its precision and efficiency. The Box-Behnken design is also a strong candidate, offering a good balance between cost and performance. In contrast, the full factorial and composite designs, while strong in terms of space coverage, are either too costly or less efficient depending on the evaluation criteria.

4.2 Comparison of Computer Design

To ensure the statistical relevance of the results, the evaluation criteria were computed over a set of 100 designs generated for each stochastic method. The comparison was based on several performance metrics, including Coverage (Cov), Discrepancy (Disc), Minimum Distance (Mindist), and the R criterion. The following types of designs were compared:

- Random Designs (RD)
- Latin Hypercube Sampling (LHS) [56]
- Maximin Latin Hypercube Sampling (mLHS) [57]
- Maximum Entropy Designs (Dmax) [44]
- Strauss Designs (SD) [36]
- Marked Strauss Designs (MSD)
- Connectivity-Interaction Model Designs (CCD) [39]
- Two-Marked Strauss Designs (TMD) [37]

4.2.1 Designs with 20, 50, and 100 Points in 5 Dimensions

The figures below provide a visual representation of the most relevant evaluation criteria computed for each design. These graphical illustrations facilitate a clearer understanding and interpretation of the results by highlighting the distribution patterns and variations observed across each criterion.

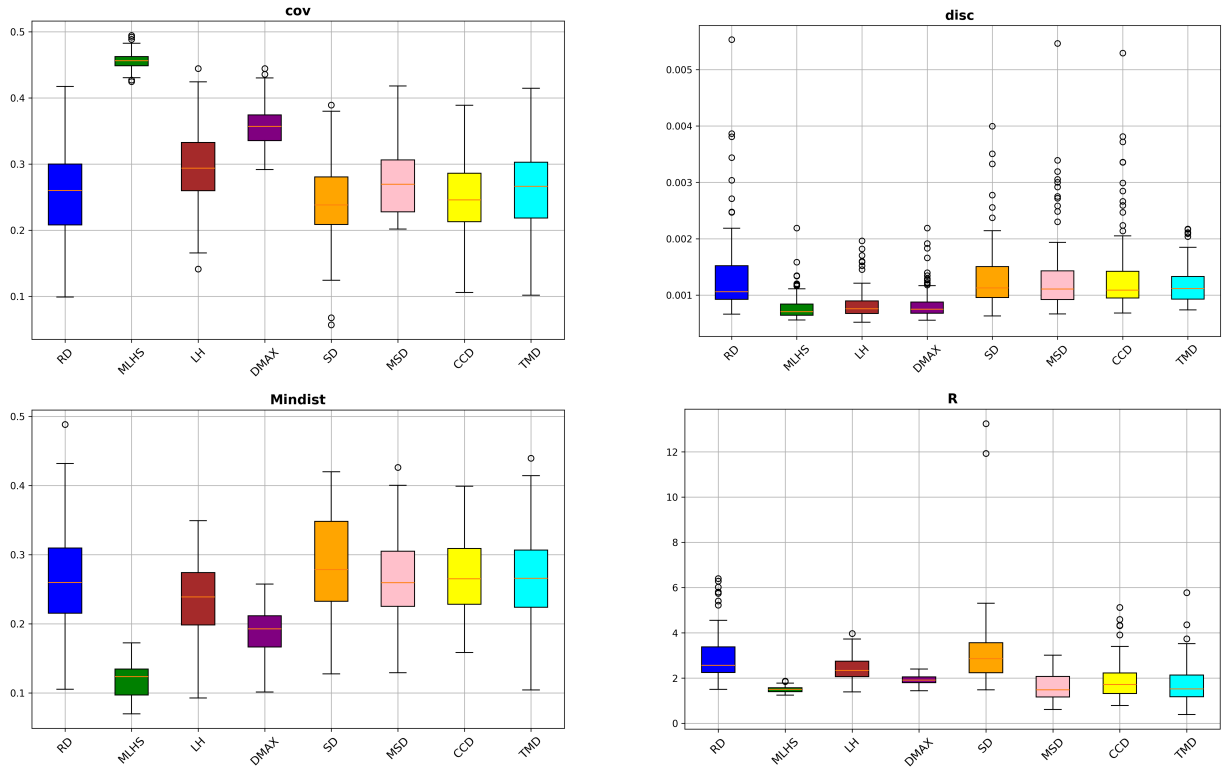

Figure 4.1: Box plots of the quality criteria computed on the 100 designs with 20 points in dimension 5.

In summary, while the optimal choice depends on the specific quality criterion being targeted, Strauss-based designs and CCD stand out as the most well-rounded and robust options for general use in computer experiments.

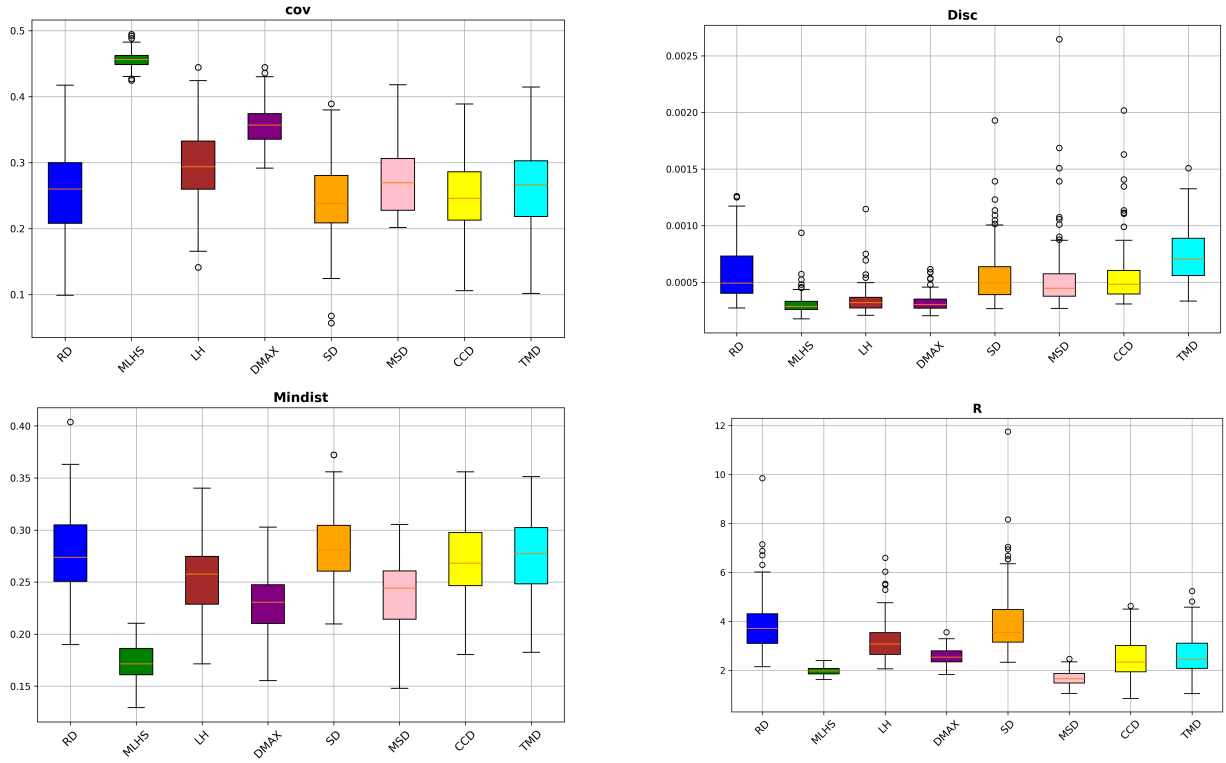

Figure 4.2: Box plots of the quality criteria computed on the 100 designs with 50 points in dimension 5.

DISCUSSION: Strauss designs (SD, MSD, TMD) and CCD exhibit the best balance between space-filling and uniformity. MLHS and Dmax achieve low discrepancy but at the cost of spacing. Random designs (RD) remain the least reliable. For dimension 5 and 50 points, MSD, CCD, and TMD offer the most robust overall quality.

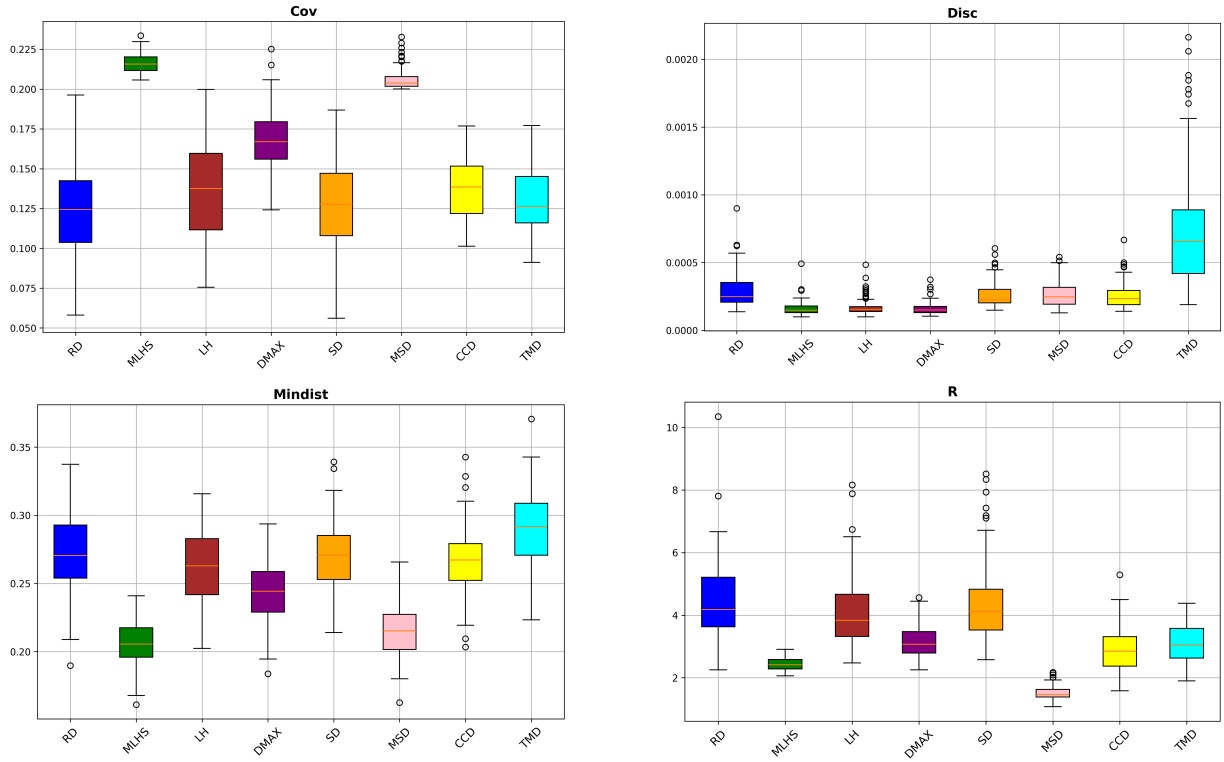

Figure 4.3: Box plots of the quality criteria computed on the 100 designs with 100 points in dimension 5.

DISCUSSION: In the case of 100 points in 5 dimensions, the best-performing designs overall are MSD, CCD, and TMD, achieving a good trade-off between space-filling (cov, mindist) and uniformity (disc). MLHS continues to perform well in uniformity but is limited by poor spacing and coverage. Random and LH designs show inconsistent results. The Strauss designs maintain good spacing, but SD can be less stable in balance (R).

4.2.2 Designs with 20, 50, and 100 Points in 7 Dimensions

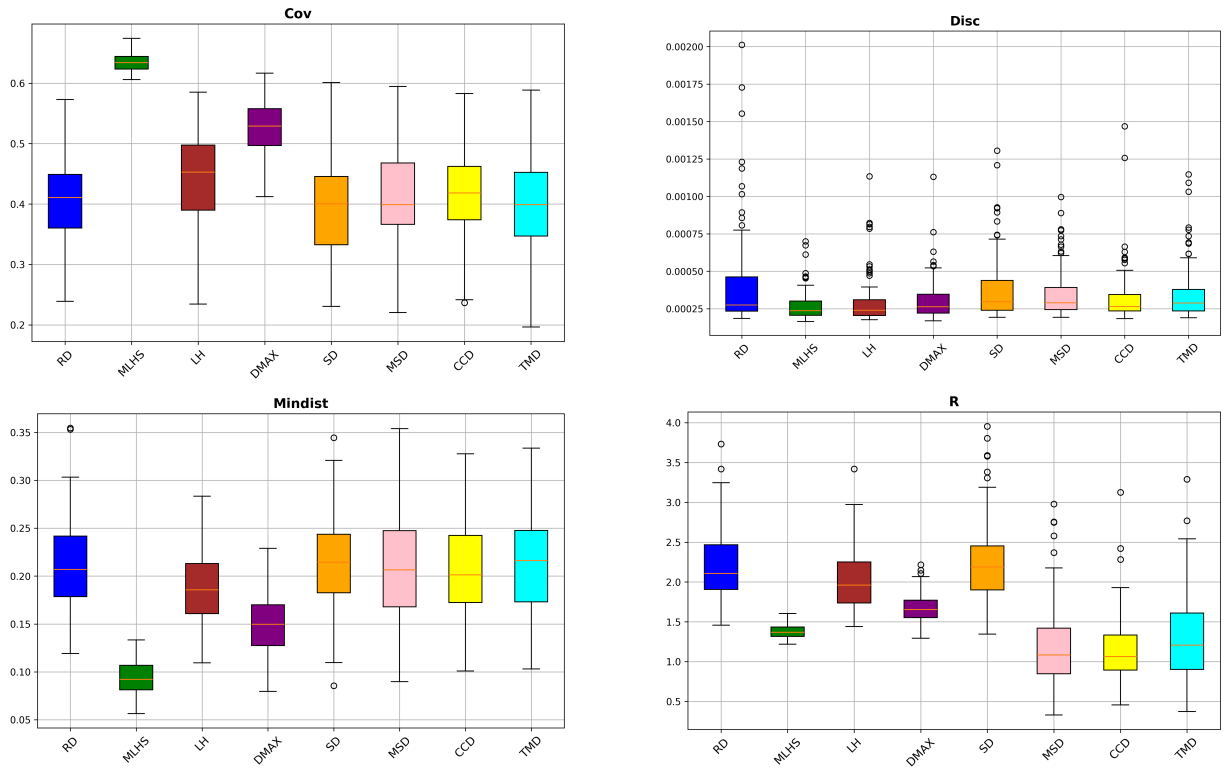

Figure 4.4: Box plots of the quality criteria computed on the 100 designs with 20 points in dimension 7.

The *Two-Marked Strauss Designs* (TMD) and *Marked Strauss Designs* (MSD) stand out with high *Mindist* values, indicating good spatial dispersion. The *Strauss-type designs* (SD, MSD) also show low *Cov* values, reflecting good alignment with a regular grid. Regarding discrepancy, the best results are obtained by the *Dmax* and *mLHS* designs, which ensure a uniform distribution of points.

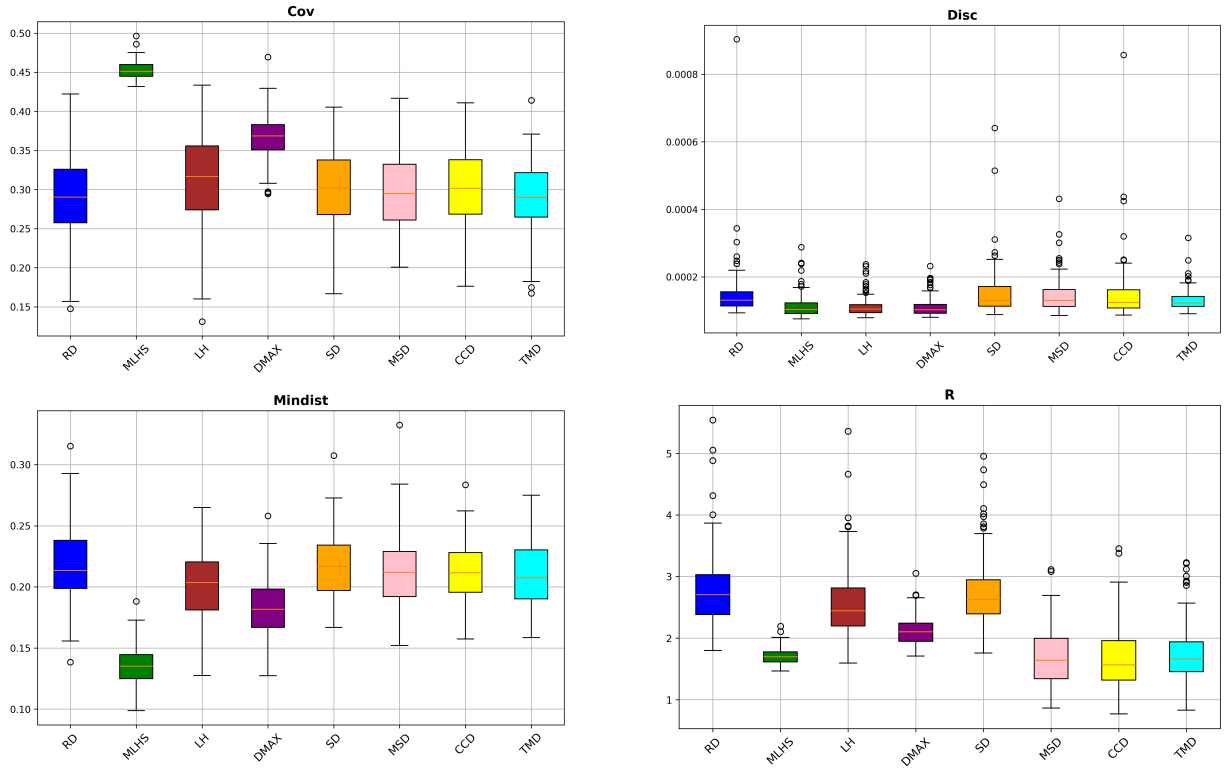

Figure 4.5: Box plots of the quality criteria computed on the 100 designs with 50 points in dimension 7.

With 50 points, the observed trends are confirmed. The TMD and MSD designs maintain their advantage in terms of minimum distance, while the SD and MSD designs continue to yield the best results for the coverage criterion. Discrepancy remains dominated by the *Dmax* designs, closely followed by *mLHS*.

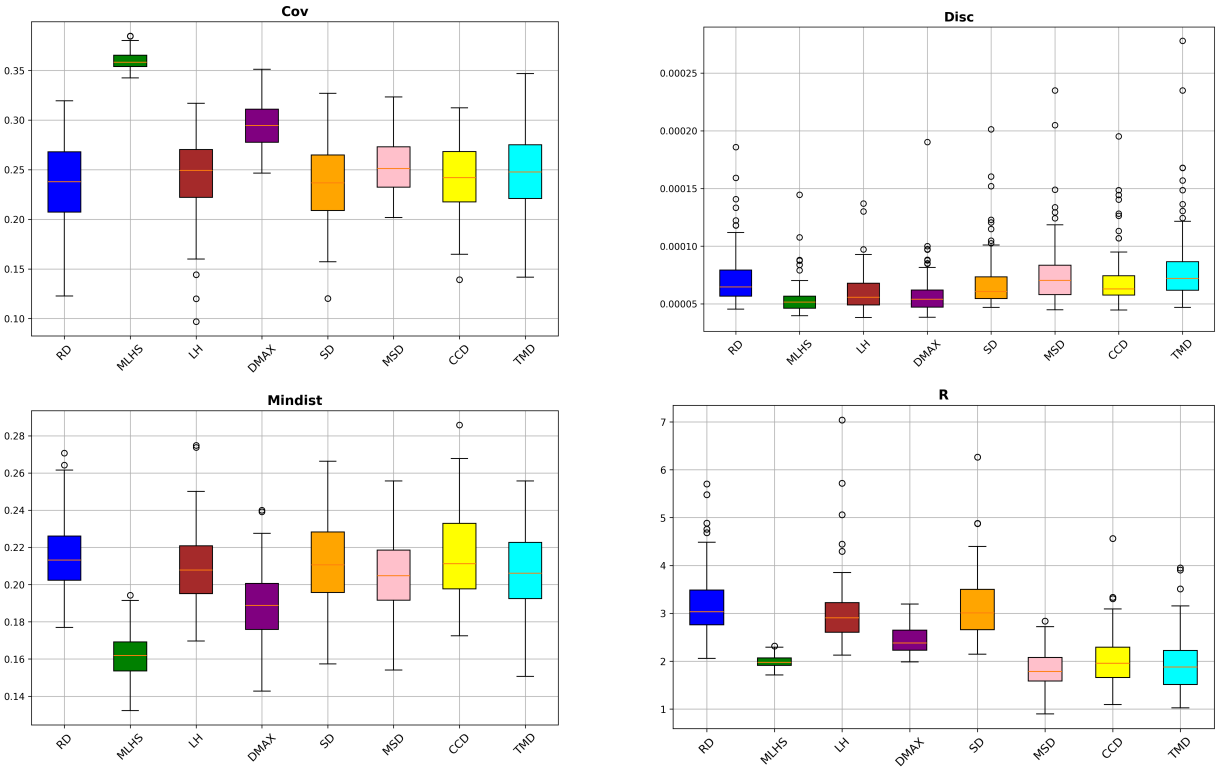

Figure 4.6: Box plots of the quality criteria computed on the 100 designs with 100 points in dimension 7.

An overall convergence in performance is observed. The differences between designs diminish, particularly for *Mindist* and *Disc*, due to the increased point density in the space. However, Dmax and mLHS designs still retain an advantage in terms of uniformity, while the Strauss-based designs maintain their robustness with respect to the coverage criterion.

In summary, TMD and MSD designs are particularly suitable when point dispersion is prioritized, whereas Dmax and mLHS designs perform better in ensuring good spatial uniformity. Strauss-based designs (SD, MSD) demonstrate remarkable stability in terms of coverage, regardless of the number of points considered.

CONCLUSION

Mathematics has long played a central role in both fundamental and applied research, producing a wide array of theoretical advances. However, the practical application of many of these advances has often lagged behind due to the computational complexity involved. The rise of computer technology has helped bridge this gap, enabling researchers to implement sophisticated models and perform large-scale computations. A prime example is the widespread adoption of statistical methods and experimental design techniques in modern industry.

This thesis presented a comprehensive synthesis of experimental design methodologies, ranging from classical standard designs to modern numerical approaches tailored for computer experiments. In the first part of the work, we reviewed the theoretical foundations and described a variety of design families, including factorial, composite, and uniform designs, as well as numerical designs derived from stochastic models, such as marked point processes.

The second part of the thesis focused on a comparative analysis of these designs based on several optimality criteria. Classical criteria such as A-, D-, E-, and G-optimality were applied to standard designs, while criteria such as discrepancy, entropy, and inter-point distance were used to assess computer-generated designs. This analysis provided insight into the strengths and limitations of each design, depending on the number of factors and the experimental budget.

The results demonstrated that standard designs are efficient and interpretable for low-dimensional problems, particularly when second-order models are required. However, in high-dimensional settings, numerical designs although more complex to construct offer better space-filling properties and prediction performance.

This work also opens several promising research directions. These include the development of new probabilistic models for design generation, the combination of multiple optimality criteria to build hybrid designs, and the incorporation of adaptive or Bayesian frameworks to dynamically optimize experiments based on intermediate results.

APPENDIX A

PYTHON code for the results of the 4th chapter

for the standard experimental designs

code for BBD with k factors

```
import numpy as np
import itertools
import scipy.linalg  # Pour une inversion plus stable
num_factors = 6
num_center_points = 3  # Nombre de points au centre
factors = list(range(num_factors))
factor_pairs = list(itertools.combinations(factors, 2))
levels = [-1, 1]
level_combinations = list(itertools.product(levels, repeat=2))
design_points = []
for pair in factor_pairs:
    for combo in level_combinations:
        run = np.zeros(num_factors)
        run[list(pair)] = combo
        design_points.append(run)
center_point = np.zeros(num_factors)
for _ in range(num_center_points):
    design_points.append(center_point)
X_design = np.array(design_points)
num_runs = X_design.shape[0]
num_linear = num_factors
num_interact = num_factors * (num_factors - 1) // 2
num_quad = num_factors
```

```
num_params = 1 + num_linear + num_interact + num_quad
X_model = np.ones((num_runs, num_params))
col_idx = 1
X_model[:, col_idx:col_idx + num_linear] = X_design
col_idx += num_linear
interaction_pairs = list(itertools.combinations(range(num_factors), 2))
for i, j in interaction_pairs:
    X_model[:, col_idx] = X_design[:, i] * X_design[:, j]
    col_idx += 1
for i in range(num_factors):
    X_model[:, col_idx] = X_design[:, i] ** 2
    col_idx += 1
Xt = X_model.T
XtX = Xt @ X_model
try:
    XtX_inv = scipy.linalg.inv(XtX)
    matrix_invertible = True
except scipy.linalg.LinAlgError:
    XtX_inv = np.full((num_params, num_params), np.nan)
    matrix_invertible = False
total_experiments = num_runs
num_levels = 3
if matrix_invertible:
    trace_XtX_inv = np.trace(XtX_inv)
    M_matrix = XtX / num_runs
    sign_M, log_det_M = np.linalg.slogdet(M_matrix)
    M_criterion = np.exp(log_det_M) if sign_M > 0 else 0
    sign, log_det_XtX = np.linalg.slogdet(XtX)
    det_XtX = np.exp(log_det_XtX) if sign > 0 else 0
    pred_var_normalized = num_runs * np.sum((X_model @ XtX_inv) * X_model,
        axis=1)
    max_pred_var_normalized = np.max(pred_var_normalized)
    g_efficiency = (num_params / max_pred_var_normalized) * 100
else:
    trace_XtX_inv = M_criterion = det_XtX = max_pred_var_normalized =
        g_efficiency = np.nan
print(f"--- Propriétés du Plan Box-Behnken k={num_factors}, nc={
    num_center_points} ---")
print(f"1. Nombre total d'expériences (N): {total_experiments}")
print(f"2. Nombre de niveaux par facteur: {num_levels}")
```

```
print("-" * 40)
print("Métriques d'Optimalité :")
print(f"3. Nombre de paramètres (p): {num_params}")
if matrix_invertible:
    print(f"4. Trace[(X^T X)^-1] (A-optimalité): {trace_XtX_inv:.6f}")
    print(f"5. Déterminant de X^T X (D-optimalité): {det_XtX:.6e}")
    print(f"6. Déterminant de M = (1/N) X^T X (M-criterion): {M_criterion
        :.6e}")
    print(f"7. G-efficiency:")
    print(f"   -> Variance de prédiction normalisée max : {
        max_pred_var_normalized:.4f}")
    print(f"   -> Efficacité G : {g_efficiency:.2f}%")
else:
    print("ATTENTION : Matrice X^T X non inversible, calculs non
        réalisables.")
print("-" * 40)
```

4.2.2.1 code for full fact 2^k

```
import numpy as np
import scipy.linalg
from pyDOE2 import ff2n
num_factors = 6  # Changer ici le nombre de facteurs
add_center_points = True
num_center_points = 0 if add_center_points else 0
X_base = ff2n(num_factors)  # Matrice codée en [-1, +1]
if add_center_points:
    center_points = np.zeros((num_center_points, num_factors))
    X_design = np.vstack((X_base, center_points))
else:
    X_design = X_base
num_runs = X_design.shape[0]
num_params = 1 + num_factors  # Intercept + termes linéaires
X_model = np.ones((num_runs, num_params))
X_model[:, 1:] = X_design  # Ajout des colonnes linéaires
XtX = X_model.T @ X_model
try:
    XtX_inv = scipy.linalg.inv(XtX)
    matrix_invertible = True
```

```
except np.linalg.LinAlgError:
    matrix_invertible = False
    XtX_inv = np.full((num_params, num_params), np.nan)
M_matrix = XtX / num_runs
sign_M, log_det_M = np.linalg.slogdet(M_matrix)
if sign_M > 0:
    M_criterion = np.exp(log_det_M)
else:
    M_criterion = 0 if sign_M == 0 else -np.exp(log_det_M)
if matrix_invertible:
    trace_XtX_inv = np.trace(XtX_inv)
    sign, log_det = np.linalg.slogdet(XtX)
    det_XtX = np.exp(log_det) if sign > 0 else 0
    pred_var = num_runs * np.sum((X_model @ XtX_inv) * X_model, axis=1)
    max_pred_var = np.max(pred_var)
    g_eff = (num_params / max_pred_var) * 100
else:
    trace_XtX_inv = det_XtX = g_eff = max_pred_var = np.nan
print(f"--- PLAN FACTORIEL 2^{num_factors} : Modèle LINÉAIRE ---")
print(f"Nombre d'expériences : {num_runs}")
print(f"Nombre de paramètres (linéaires) : {num_params}")
print("-" * 40)
if matrix_invertible:
    print(f"Trace[(X^T X)^-1] (A-optimalité) : {trace_XtX_inv:.6f}")
    print(f"Déterminant X^T X (D-optimalité) : {det_XtX:.6e}")
    print(f"M-criterion (det[(1/N) X^T X]) : {M_criterion:.6e}")
    print(f"-> Interprétation : Plus cest grand, plus le plan est
       informatif (normalisé à N).")

    print(f"Max variance de prédiction : {max_pred_var:.6f}")
    print(f"G-efficiency : {g_eff:.2f}%")
else:
    print("ATTENTION : Matrice X^T X non inversible.")
print("-" * 40)
```

4.2.2.2 code for full fact 3^k

```
import numpy as np
import itertools
```

```
import scipy.linalg # Pour une inversion potentiellement plus stable
from pyDOE2 import fullfact
num_factors = 7
levels = [3] * num_factors
X_design_012 = fullfact(levels)
X_design = X_design_012 - 1
num_runs = X_design.shape[0]
num_runs = X_design.shape[0]# Nombre total d'expériences
print("Niveaux uniques dans le CCD:", np.unique(X_design))
num_linear = num_factors
num_interact = num_factors * (num_factors - 1) // 2
num_quad = num_factors
num_params = 1 + num_linear + num_interact + num_quad
X_model = np.ones((num_runs, num_params))
col_idx = 1
X_model[:, col_idx : col_idx + num_linear] = X_design
col_idx += num_linear
interaction_pairs = list(itertools.combinations(range(num_factors), 2))
for i, j in interaction_pairs:
    X_model[:, col_idx] = X_design[:, i] * X_design[:, j]
    col_idx += 1
for i in range(num_factors):
    X_model[:, col_idx] = X_design[:, i] ** 2
    col_idx += 1
Xt = X_model.T
XtX = Xt @ X_model  # Moment matrix M = X^T X
M_matrix = XtX / num_runs
sign_M, log_det_M = np.linalg.slogdet(M_matrix)
if sign_M > 0:
    M_criterion = np.exp(log_det_M)
else:
   M_criterion = 0 if sign_M == 0 else -np.exp(log_det_M)
try:
    XtX_inv = scipy.linalg.inv(XtX)
    matrix_invertible = True
except scipy.linalg.LinAlgError:
    print("Erreur: La matrice singulière, impossible de calculel'inverse.")
    matrix_invertible = False
    XtX_inv = np.full((num_params, num_params), np.nan)
total_experiments = num_runs
```

```
num_levels = 3 # Par définition du plan Box-Behnken (-1, 0, +1)
if matrix_invertible:
    trace_XtX_inv = np.trace(XtX_inv)
else:
    trace_XtX_inv = np.nan
sign, log_det_XtX = np.linalg.slogdet(XtX)
if sign > 0:
    det_XtX = np.exp(log_det_XtX) # det = exp(log(det))
else:
    det_XtX = 0 if sign == 0 else -np.exp(log_det_XtX)
if matrix_invertible:
    pred_var_normalized = num_runs * np.sum((X_model @ XtX_inv) * X_model,
        axis=1)
    max_pred_var_normalized = np.max(pred_var_normalized)
    g_efficiency = (num_params / max_pred_var_normalized) * 100
else:
    max_pred_var_normalized = np.nan
    g_efficiency = np.nan
print(f"1. Nombre total d'expériences (N): {total_experiments}")
print(f"2. Nombre de niveaux par facteur: {num_levels}")
print("-" * 40)
print("Métriques d'Optimalité (basées sur le modèle quadratique) :")
print(f"   Nombre de paramètres dans le modèle (p): {num_params}")
if matrix_invertible:
    print(f"3. Trace[(XT X)1] (A-optimality related): {trace_XtX_inv:.6f}")
    print(f"   -> Interprétation: Plus c'est petit, meilleure est la
        variance moyenne des estimations.")
    print(f"4. Déterminant de M = XT X (D-optimality related): {det_XtX:.6e
        }")
    print(f"   -> Interprétation: Plus c'est grand, plus le volume de
        confiance des paramètres est petit.")
    print(f"5. M-criterion (Determinant de M = (1/N) X^T X): {M_criterion
        :.6e}")
    print(f"   -> Interprétation: Plus c'est grand, plus l'information
        totale est concentrée et fiable.")
    print(f"6. G-efficiency:")
    print(f"   -> Variance de prédiction normalisée max (sur les points du
        plan): {max_pred_var_normalized:.4f}")
    print(f"   -> Efficacité G = (p / max_var_norm) * 100: {g_efficiency:.2
        f}%")
```

```
    print(f"   -> Interprétation: Proche de 100% indique une variance de
        prédiction uniforme sur les points du plan.")
else:
    print("3. Trace[(XT X)1]: Non calculable (matrice XT X singulière)")
    print("4. Déterminant de M = XT X: Non calculable (matrice XT X
        singulière ou non définie positive)")
    print("5. G-efficiency: Non calculable (matrice XT X singulière)")
print("-" * 40)
np.set_printoptions()
```

```
import numpy as np
import itertools
import scipy.linalg
num_factors = 6
num_center_points = 3 # Ajout de points centraux
def generate_mozzo(k):
    points = []
    for i in range(k):
        pt = np.zeros(k)
        pt[i] = 1
        points.append(pt)
    for p in range(2, k+1):
        for comb in itertools.combinations(range(k), p):
            pt = np.zeros(k)
            pt[list(comb)] = 1/p
            points.append(pt)
    return np.array(points)
X_mozzo_base = generate_mozzo(num_factors)
X_mozzo_base = X_mozzo_base - 1/num_factors
center_points = np.zeros((num_center_points, num_factors))
X_design = np.vstack((X_mozzo_base, center_points))
num_runs = X_design.shape[0]
print(f"Plan de Mozzo (k={num_factors}, nc={num_center_points}) généré.")
print(f"Nombre total d'essais (N): {num_runs}")
num_linear = num_factors
num_interact = num_factors * (num_factors - 1) // 2
num_quad = num_factors
num_params = 1 + num_linear + num_interact + num_quad
X_model = np.ones((num_runs, num_params))
col_idx = 1
```

```
X_model[:, col_idx : col_idx + num_linear] = X_design
col_idx += num_linear
interaction_pairs = list(itertools.combinations(range(num_factors), 2))
for i, j in interaction_pairs:
    X_model[:, col_idx] = X_design[:, i] * X_design[:, j]
    col_idx += 1
for i in range(num_factors):
    X_model[:, col_idx] = X_design[:, i] ** 2
    col_idx += 1
Xt = X_model.T
XtX = Xt @ X_model
try:
    XtX_inv = scipy.linalg.inv(XtX)
    matrix_invertible = True
except scipy.linalg.LinAlgError:
    matrix_invertible = False
    XtX_inv = np.full((num_params, num_params), np.nan)
total_experiments = num_runs
num_levels = len(np.unique(X_design))
if matrix_invertible:
    trace_XtX_inv = np.trace(XtX_inv)
    sign, log_det_XtX = np.linalg.slogdet(XtX)
    det_XtX = np.exp(log_det_XtX) if sign > 0 else (0 if sign == 0 else -np
        .exp(log_det_XtX))
    M_matrix = XtX / num_runs
    sign_M, log_det_M = np.linalg.slogdet(M_matrix)
    M_criterion = np.exp(log_det_M) if sign_M > 0 else (0 if sign_M == 0
        else -np.exp(log_det_M))
    pred_var_normalized = num_runs * np.sum((X_model @ XtX_inv) * X_model,
        axis=1)
    max_pred_var_normalized = np.max(pred_var_normalized)
    g_efficiency = (num_params / max_pred_var_normalized) * 100
else:
    trace_XtX_inv = np.nan
    det_XtX = np.nan
    g_efficiency = np.nan
print("-" * 50)
print(f"--- Propriétés du Plan de Mozzo k={num_factors}, nc={
    num_center_points} ---")
print(f"1. Nombre total d'expériences (N): {total_experiments}")
```

```
print(f"2. Nombre de niveaux uniques par facteur: {num_levels}")
print("-" * 40)
print("Métriques d'Optimalité (basées sur le modèle quadratique) :")
print(f"   Nombre de paramètres dans le modèle (p): {num_params}")
if matrix_invertible:
    print(f"3. Trace[(XT X)1] (A-optimalité): {trace_XtX_inv:.6f}")
    print(f"4. Déterminant de M = XT X (D-optimalité): {det_XtX:.6e}")
    print(f"5. M-criterion = det[(1/N) * X^T X] : {M_criterion:.6e}")
    print(f"6. G-efficiency : {g_efficiency:.2f}%")
    print(f"   -> Max variance de prédiction normalisée: {
       max_pred_var_normalized:.4f}")
else:
    print("Matrice XT X non inversible : métriques non calculables.")
print("-" * 50)
```

BIBLIOGRAPHY

[1] P. Schimmerling, J. C. Sisson, A. Zaïdi, Pratique des plans d'expériences, TEC & DOC, 1998.

[2] W. Tinsson, Plans d'expérience: constructions et analyses statistiques, Vol. 67, Springer Science & Business Media, 2010.

[3] A. W. Edwards, Ra fischer, statistical methods for research workers, (1925), in: Landmark writings in western mathematics 1640-1940, Elsevier, 2005, pp. 856–870.

[4] W. G. Hunter, J. S. Hunter, G. Box, Statistics for experimenters, Interscience, New York 453 (1978).

[5] G. Taguchi, Introduction to quality engineering: designing quality into products and processes, 1986.

[6] J. A. Cornell, Experiments with mixtures: designs, models, and the analysis of mixture data, Vol. 255, Wiley-interscience, 1843.

[7] E. Williams, Cyclic and computer generated designs, Chapman & Hall, 1995.

[8] G. A. Seber, C. J. Wild, Nonlinear regression. hoboken, New Jersey: John Wiley & Sons 62 (63) (2003) 1238.

[9] T. J. Santner, B. J. Williams, W. I. Notz, B. J. Williams, The design and analysis of computer experiments, Vol. 1, Springer, 2003.

[10] J. Goupy, Modélisation par les plans d'expériences, Techniques de l'ingénieur. Mesures et contrôle (R275) (2000) R275–1.

[11] D. C. Montgomery, Design and analysis of experiments, John wiley & sons, 2017.

[12] G. E. Box, N. R. Draper, Response surfaces, Mixtures, and Ridge Analyses 649 (2007).

[13] J. Goupy, Plans d'expériences pour surfaces de réponse, Dunod, 1999.

[14] R. H. Myers, D. C. Montgomery, C. M. Anderson-Cook, Response surface methodology: process and product optimization using designed experiments, John Wiley & Sons, 2016.

[15] G. Saporta, Probabilités, analyse des données et statistique, Editions technip, 2007.

[16] J. Goupy, La méthode des plans d'expériences: optimisation du choix des essais & de l'interprétation des résultats, Vol. 29, Dunod Paris, 1988.

[17] H. Ahrens, Searle, sr: Linear models. john wiley & sons, inc., new york-london-sydney-toronto 1971. xxi, 532 s. $9.50 (1974).

[18] Y. Dodge, V. Rousson, Analyse de régression appliquée, Dunod, 1999.

[19] R. H. Myers, D. C. Montgomery, Response surface methodology (1996).

[20] G. E. Box, D. W. Behnken, Some new three level designs for the study of quantitative variables, Technometrics 2 (4) (1960) 455–475.

[21] J.-J. Droesbeke, J. Fine, G. Saporta, Plans d'expériences: applications à l'entreprise, Editions technip, 1997.

[22] G. Mozzo, Plans dexpériences séquentiel à symétrie évolutive, Tech. rep., Elf-AtoChem, document interne, 18 pages (1995).

[23] A. B. Owen, Orthogonal arrays for computer experiments, integration and visualization, Statistica Sinica (1992) 439–452.

[24] A. Jourdan-Marias, Analyse statistique et échantillonage d'expériences simulées, Ph.D. thesis, Pau (2000).

[25] B. Tang, Orthogonal array-based latin hypercubes, Journal of the American statistical association 88 (424) (1993) 1392–1397.

[26] E. Thiémard, Sur le calcul et la majoration de la discrépance à l'origine, Tech. rep., EPFL (2000).

[27] J. H. Halton, On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals, Numerische Mathematik 2 (1960) 84–90.

[28] J. M. Hammersley, Monte carlo methods for solving multivariable problems, Annals of the New York Academy of Sciences 86 (3) (1960) 844–874.

[29] I. M. Sobol, The distribution of points in a cube and the approximate evaluation of integrals, USSR Computational mathematics and mathematical physics 7 (1967) 86–112.

[30] H. Faure, Discrépance de suites associées à un système de numération (en dimension s), Acta arithmetica 41 (4) (1982) 337–351.

[31] H. Niederreiter, Low-discrepancy and low-dispersion sequences, Journal of number theory 30 (1) (1988) 51–70.

[32] H. Faure, Suitesa faible discrépance dans ts, Publ. Dép. Math., Université de Limoges, Limoges, France (1980).

[33] H. Niederreiter, Random number generation and quasi-Monte Carlo methods, SIAM, 1992.

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

[35] S. Joe, F. Y. Kuo, Remark on algorithm 659: Implementing sobol's quasirandom sequence generator, ACM Transactions on Mathematical Software (TOMS) 29 (1) (2003) 49–57.

[36] J. Franco, X. Bay, D. Dupuy, B. Corre, Planification d'expériences numériques à partir du processus ponctuel de strauss (2008).

[37] H. Elmossaoui, N. Oukid, F. Hannane, Construction of computer experiment designs using marked point processes, Afrika Matematika 31 (2020) 917–928.

[38] H. Elmossaoui, A. Ait Ameur, N. Oukid, Improving computer experiment designs with two-type marked point processes, International Journal of Analysis and Applications 23 (2025) 82.

[39] H. Elmossaoui, N. Oukid, New computer experiment designs using continuum random cluster point process, International Journal of Analysis and Applications 21 (2023) 51–51.

[40] A. Ait Ameur, H. Elmossaoui, N. Oukid, New computer experiment designs with area-interaction point processes, Mathematics 12 (15) (2024) 2397.

[41] J. Dick, F. Pillichshammer, Digital nets and sequences: discrepancy theory and quasi–Monte Carlo integration, Cambridge University Press, 2010.

[42] F. Hickernell, A generalized discrepancy and quadrature error bound, Mathematics of computation 67 (221) (1998) 299–322.

[43] V. R. Joseph, Y. Hung, A. Sudjianto, Blind kriging: A new method for developing metamodels, Mechanical Design (2008).

[44] M. C. Shewry, H. P. Wynn, Maximum entropy sampling, Journal of applied statistics 14 (2) (1987) 165–170.

[45] K. Chaloner, I. Verdinelli, Bayesian experimental design: A review, Statistical science (1995) 273–304.

[46] K.-T. Fang, R. Li, A. Sudjianto, Design and modeling for computer experiments, Chapman and Hall/CRC, 2005.

[47] C. Lemieux, Quasi–monte carlo constructions, in: Monte Carlo and Quasi-Monte Carlo Sampling, Springer, 2008, pp. 1–61.

[48] M. E. Johnson, L. M. Moore, D. Ylvisaker, Minimax and maximin distance designs, Journal of statistical planning and inference 26 (2) (1990) 131–148.

[49] H. Faure, Discrépances de suites associées à un système de numération (en dimension un), Bulletin de la Société Mathématique de France 109 (1981) 143–182.

[50] M. Gunzburger, J. Burkardt, Uniformity measures for point sample in hypercubes, Rapp. tech. Florida State University (cf. p. 73) (2004).

[51] V. C. Chen, K.-L. Tsui, R. R. Barton, J. K. Allen, Ch. 7. a review of design and modeling in computer experiments, Handbook of statistics 22 (2003) 231–261.

[52] J. R. Koehler, A. B. Owen, 9 computer experiments, Handbook of statistics 13 (1996) 261–308.

[53] T. J. Mitchell, D. S. Scott, A computer program for the design of group testing experiments, Communications in Statistics-Theory and methods 16 (10) (1987) 2943–2955.

[54] C. Currin, T. Mitchell, M. Morris, D. Ylvisaker, Bayesian prediction of deterministic functions, with applications to the design and analysis of computer experiments, Journal of the American Statistical Association 86 (416) (1991) 953–963.

[55] P. Sebastiani, H. P. Wynn, Maximum entropy sampling and optimal bayesian experimental design, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 62 (1) (2000) 145–157.

[56] W.-L. Loh, On latin hypercube sampling, The annals of statistics 24 (5) (1996) 2058–2080.

[57] M. D. Morris, T. J. Mitchell, Exploratory designs for computational experiments, Journal of statistical planning and inference 43 (3) (1995) 381–402.