الجمهورية الجزائرية الديمقراطية الشعبية

People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University of Saad Dahleb BLIDA 1
Faculty of Sciences

Department of Mathematics

Master Thesis

Option: *Stochastic and statistical modeling*

Type of mixture designs and optimization according to the D-optimal criterion using an exchange algorithm

Presented by :

BEN OKBA Yassine and LAZGHED Nouredddine

supervisor:

Dr ELMOSSAOUI H. Associate Professor A

examing boord:

Dr MESSAOUDI A.N.	Associate Professor B	President
Mr BOUKHARI M.	Assistant Professor A	Examiner

2023/ 2024

Abstract

Most engineers and technicians seek to improve their products or production processes through experimentation. Unfortunately, the strategies commonly used to conduct these experiments are often costly, inefficient, and result in numerous difficult-to-exploit experiments. For these reasons, many engineers and technicians turn to experimental design, particularly Mixture Experimental Designs.

In this dissertation, a detailed study of mixture designs was given, and an software application was developed by python programming language to handle mixture designs. This provides a good understanding of the contribution of computer science to experimental design and the assistance an experimenter can expect from it.

Key Words : Experimental designs, mixture designs, optimality criteria, D-optimality criterion, exchange algorithms.

ملخص

يطمح معظم المهندسين والفنيين باستمرار إلى تحسين منتجاتهم أو عمليات الإنتاج من خلال التجارب. لكن للأسف، غالبًا ما تكون الاستراتيجيات الشائعة المستخدمة لإجراء هذه التجارب مكلفة وغير فعالة وتؤدي إلى العديد من التجارب الصعبة التطبيق.

ولذلك يلجأ العديد من المهندسين والفنيين إلى تصميم التجارب، وخاصة تصاميم خلط المكونات.

تُقدم هذه الأطروحة دراسة تفصيلية لتصميمات خلط المكونات، كما تم تطوير تطبيق بايثون للتعامل مع هذه التصميمات. يوفر هذا التطبيق فهماً جيداً لمساهمة علوم الكمبيوتر في تصميم التجارب والمساعدة التي يمكن أن يتوقعها المجرب منه.

الكلمات المفتاحية: التصاميم التجريبية، تصاميم الخلطات، معايير الأمثلية، معيار الأمثلية D، خوارزميات التبادل.

Résumé

La plupart des ingénieurs et techniciens cherchent à améliorer leurs produits ou leurs processus de production par l'expérimentation. Malheureusement, les stratégies couramment utilisées pour mener ces

les expériences sont souvent coûteuses, inefficaces et donnent lieu à de nombreuses expériences difficiles à exploiter. Pour ces raisons, de nombreux ingénieurs et techniciens se tournent vers la conception expérimentale, notamment Modèles expérimentaux de mélanges.

Dans cette thèse, une étude détaillée des conceptions de mélanges a été menée et un Python L'application a été développée pour gérer les conceptions de mélanges. Cela permet de bien comprendre l'apport de l'informatique à la conception expérimentale et l'assistance d'un expérimentateur peut en attendre

Mots clés: Plans expérimentaux, plans de mélange, critères d'optimalité, critère D-optimalité, algorithmes d'échange.

Acknowledgements

We thank Allah, the Almighty, for the will, patience, and courage that He has granted us to complete this work. We would like to express our sincere appreciation to Dr. ELMOSSAOUI Hichem for agreeing to supervise us, for his discreet attention, measured recommendations, and valuable advice, and especially for his human and scientific qualities always in modesty, his passion for the profession which he knows how to make contagious, and the trust he has been willing to grant us throughout this work.

Next, we would like to thank all the members of THE JURY who will evaluate our work and provide us with their knowledge and constructive feedback. We want to express our gratitude to all the teachers of the mathematics department, especially to the department head Dr. TAMI Omar.

Dedication

To the one who has always welcomed me with open arms and supported me in all my endeavors, the one who has been kind, gentle, and understanding with me, the one whose simple smile lights up my life, my precious and dear **mother**;

To the one who taught me that patience is the key to success and victory, the one who has greatly contributed to my development, the one who has always encouraged me to move forward, the one who always saw a winner in me and never doubted me, my dear and revered **Father**;

To my **aunt**, who has been a constant source of inspiration and guidance, always showering me with tenderness and support, sharing her wisdom and unconditional love with me; To my beloved **grandma and grand aunt and granduncle** , whose warmth and wisdom have illuminated my path, providing strength and comfort in times of need, their memory will live on in the hearts of those who knew and loved them May Allah have mercy on them; To my dear **little aunt** , whose joyful presence and generous spirit have filled our family gatherings with laughter and kindness; To my two cherished **cousins**, who have stood by me through both triumphs and challenges, their friendship a source of joy and support throughout the years.

To my dear **younger sisters**, In your laughter, I find echoes of our shared memories, each moment a treasure in the tapestry of our lives. May the path ahead be lined with endless possibilities, and may your spirits forever dance in the light of hope and fulfillment.

And to my **classmates** , whose camaraderie and shared experiences have made our academic journey memorable and enriching; to my **roommates** , whose companionship and understanding have turned our shared living space into a home away from home; and to my **neighbors** , whose friendly faces and sense of community have made our surroundings a welcoming and vibrant place to be.

May God protect you all, grant you health, prosperity, and a long life.

Dedication

This work is dedicated to those who have fueled my journey:

***My mother**, whose unwavering support and encouragement were the fuel that propelled me through my studies.*

***Dad**, your belief in me from the very beginning empowered me to reach this point.*

*To my two big brothers, **Nadjib** and especially **Abdelwahab**, my rocks, I wouldn't be here without you.*

*To my amazing **sisters**, your love and constant encouragement were invaluable.*

*And to my incredible **life partner**, your unwavering support made all the difference*

This work serves as a token of my deepest gratitude to all of you.

*Additionally, I extend my thanks to my colleagues in work and everyone who knows and supports **Lazghed Noureddine**.*

Your well wishes have not gone unnoticed.

Contents

List of Figures

List of Tables

Introduction

The Methodology of Experimental Research (Design of Experiments Method) is useful to all those who undertake scientific research or industrial studies. The use of experimental designs for the empirical study of a response law poses particular problems for statisticians or researchers. While they have little information about this law, they generally only have a very limited sample of observations compared to the number of parameters of the models they can consider for their analyses. Before any observation of the response, they must therefore specify not only which models to use but also how to organize the experiments. Indeed, the quality of the statistical analysis closely depends on the experimental design used to observe the response. Furthermore, it is generally necessary to resort to combinatorial analysis to construct the proposed experimental designs. To propose a solution that meets industrial objectives, it is sometimes necessary to seek the missing information by conducting a set of experiments.

Important decisions made based on experimental results and the significant cost of experimentation prohibit leaving the search for the solution to the problem solely to the experimenter's intuition. It is necessary to use a methodological approach that not only reduces the cost of experimen 9alktation but also establishes an optimal organization of the experiments. The purpose of the design of experiments method is to propose one or several strategies to solve a particular problem posed by experimental research. In our work, the general principles of constructing mixture experimental designs are presented based on the notion of experimental space. The geometric representation of experimental points is very informative, but it is quickly limited as the dimension of the space increases. This is why the matrix representation is used. With the help of both geometric and matrix representations, the main mixture experimental designs are described: Type I Mixture Designs, Type II Mixture Designs, and Type III Mixture Designs.

An application by using PYTHON has been established. It allows the construction of the three designs mentioned above, provides the model, as well as the analysis and numerical results enabling the successful validation of the mathematical model.

The thesis is composed of four chapters organized as follows:

The first chapter is devoted to the description of three mixture experimental designs that were the subject of our application.

The second chapter is dedicated to the study of the D-optimality criterion, which is widely used in experimentation to obtain a reliable prediction model.

The third chapter presents the different exchange methods to determine a subset of experimental points from a given set based on the D-optimality criterion.

Finally, the description of the developed application is discussed in chapter four.

Chapter 1

General notions on Mixture design (Type I, Type II, Type III):

Mixture designs play a crucial role in experimental design when the variables x_i represent elements of a blend or mixture. The requirement that the mixture's component parts total up to 1 or 100 % applies to these designs, creating a distinct design area that differs from traditional design settings [Mitchell (1974)].

In domains like food science, where one may wish to optimize a recipe for a specific food product, these designs are invaluable. For instance, when creating a new line of dry cake combinations, the amounts of modified cornstarch, sugar, malt dextrin, salt, sorbitol, and emulsifiers may be critical factors.

In this chapter, we will introduce the three types of mixture experimental designs along with examples to illustrate their applications. Each design type offers unique advantages and considerations, which we will explore in detail.

Producing and evaluating various formulations with different ratios of these components is the primary objective of mixture designs. Following preparation, a panel of experienced sensory analysts and physicochemical experts can assess the goods to determine their quality and effectiveness.

mixture designs offer a unique testing environment that must be carefully considered when designing experiments. Unlike traditional design possibilities, they focus on blending components that cannot be changed separately, resulting in a constrained design space.

1.1 History

With a limitless number of conceivable combinations, planning excellent mixing tests was a challenge. Limited data and statistical methodologies were accessible to researchers. Still, with better experimental designs and statistical analysis over the past few decades, improvement has been achieved. There is continuous research being done to determine priority mixes and power trials properly. In order to improve formulations incorporating many components, the idea of mixture design was developed. The first known occurrence of it dates back to [Lehmann and Scheffé (1950)] works, where he introduced network architectures for polynomial models. His studies serve as the foundation for further works, such as those of [Lambrakis (1968)], who expanded Scheffé networks in numerous ways. However, he most recently established the ultimate regulations in [Cornell (1843)] book.

1.2 Definition

A special kind of experimental design called mixture design is applied when the elements are parts of a blend or mixture. This design uses the proportions of several mixed components as independent variables. It is considered that the reaction in these tests is only dependent on the relative amounts of the mixture's elements or components, not on the mixture's total volume.

1.3 Graphic representation of a mixture

1.3.1 Representation on a line segment if we have two components

The line segment can be understood as a mixture composition in the context of mixture design. Assume that we have a mixture that consists of two components, A and B. Pure components are represented by the endpoints A and B. All potential combinations of these elements are represented by the line segment that runs between A and B.

We can experiment with different component proportions in the mixture by changing the parameter (M).

For instance, we obtain a 50% mixing of components A and B when (M = 0.5).

Figure 1.1: Mixture representation for two constituents

Every point M in the AB segment represents a binary mixture.

In A, the constituent A is the only one in the mixture.

In B, the constituent B is the only one in the mixture.

In M, $x1$ corresponds to A's proposal and $x2$ corresponds to:

$$x_1 = \frac{MB}{AB} \quad \text{et} \quad x_2 = \frac{MA}{AB}$$

1.3.2 Representation inside an equilateral triangle if we have three components

The combination is ternary if $k = 3$. The figure indicates the variance in the mixture's composition. The equilateral triangle's characteristics guarantee that, for every point M inside the triangle, the total of the lengths $Mx_1 + Mx_2 + Mx_3$ equals the length of the triangle's side. A ternary mixture can be represented by any point in the triangle ABC. As with the other elements, pure constituent A will be represented at vertex A. The mixtures without component A will be depicted on side BC. Point M represents the combination made up of $x_1 = 0.333$ of A, $x_2 = 0.167$of B, and $x_3 = 0.5$ of C.

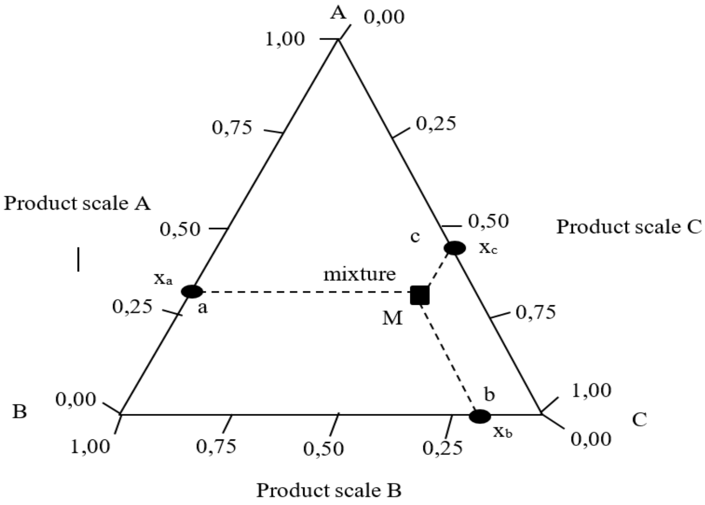


Figure 1.2: Mixture representation for three constituents

1.3.3 Representation inside a regular tetrahedron if we have four components

The mixture is quaternaire if $k = 4$. A point within a regular tétraèdre is used to represent it in the figure 1.3.

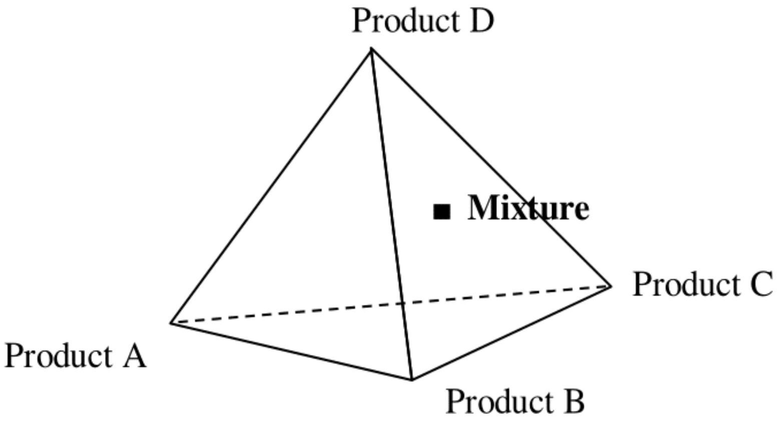


Figure 1.3: Mixture representation for four constituents

1.3.4 Representation inside a regular polyhedron if we have more then four components

This hyperpolihedron is built in a space that is k - 1.

1.4 Mixture model

The vector of proportions of the k mixture components is represented by $x = (x_1, x_2, ..., x_k)$, and the accompanying mean response is denoted by $n(x)$. In factor space notation, a simplex is represented by.

$$x = (x_1, ..., x_k) : x_i \geq 0, i = 1, 2, ..., k;\ and\ \sum_{i=1}^{k} x_i = 1$$

Scheffé (1958) introduced the canonical forms of several models to show the $n(x)$ mean response function.

$$\begin{aligned}
\text{Linear :} n(x) &= \sum_{i=1}^{q} \beta_i x_i \\
\text{Quadratic :} n(x) &= \sum_{i=1}^{k} \beta_i x_i + \sum_{i<j}^{k} \beta_{ij} x_i x_j \\
\text{Cubic :} n(x) &= \sum_{i=1}^{k} \beta_i x_i + \sum_{i<j}^{k} \beta_{ij} x_i x_j + \sum_{i<j<q}^{k} \beta_{ijq} x_i x_j x_q \\
\text{Special Cubic :} n(x) &= \sum_{i=1}^{k} \beta_i x_i + \sum_{i<j}^{k} \beta_{ij} x_i x_j + \sum_{i<j<q}^{k} \beta_{ijq} x_i x_j x_q + \sum_{i<j} \beta_{ij} x_i x_j (x_i - x_j)
\end{aligned}$$

1.5 Different types of mixture model

Mixture designs are essentially design layouts that indicate the masses corresponding to different design points or vectors of type x of mixing proportions inside the simplex. A mixed design is somewhat arbitrary in terms of the design points and their mass distribution. Generally, we assign a mass distribution after evaluating a set of design points. A collection of design points is also known, roughly speaking, as a mixing design. All of the collection's design points have been given positive masses, and it is implicitly understood that the underlying mass distribution will be determined later. [Scheffé (1958)(1963)]introduced standard mixing designs in this respect. As stated by [Sinha (2014)], Several scientists, including Cornell and Cavier, followed suit with designs that contain vertex points. Among the most used types:

1.5.1 Type I mixture design: $0 \leq x_i \leq 1$

Type I plans are such that:

$$\sum_{i=1}^{k} x_i = 1$$

1.5.1.1 Scheffé networks

Scheffé observed that while taking combinations with k elements, we should take the x_i in the series

$$0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n}{n}$$

with integer n, there are as many combinations, as many mixtures different than coefficients in the polynomial model of degree n with k constituents. The many combinations that are so created make up the Scheffé network k, n. The graphic 1.4 shows a few common networks that we have provided.

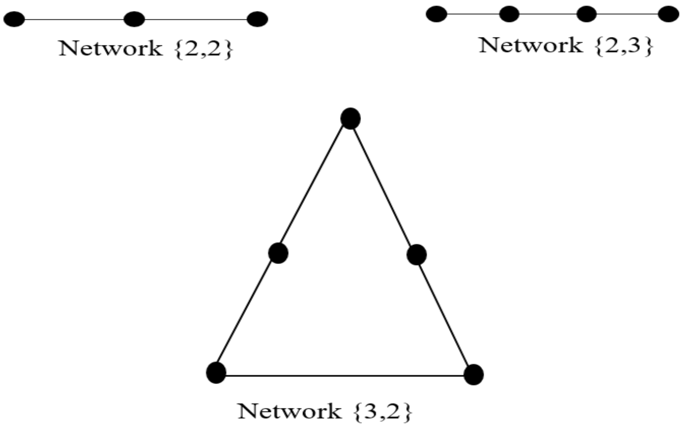


Figure 1.4: Mixture representation for two constituents

1.5.1.2 The approach

The approach used in mixture designs is as follows:

- Creating a first-degree model hypothesis, executing the network Scheffé $\{k,\ 1\}$ mixtures, and computing the model's coefficients.
- Making one or more mixes inside the domain will allow you to test the model's validity. Problem solved if validity is acknowledged,
- If the validity is rejected, formulate the second-degree model hypothesis, execute the mixtures to finish the network $\{k, 2\}$, and determine the model's coefficients. This is how validity is tested.

1.5.1.3 .Example

The behavior of a combination of three components at low temperatures is of interest to us. The cold resistance precision is ±0.5°C.

1.5.1.3.1 a) Linear Model Hypothesis: is:

$$y = a_1x_1 + a_2x_2 + a_3x_3$$

Points 1, 2, and 3 are part of the Scheffé network.

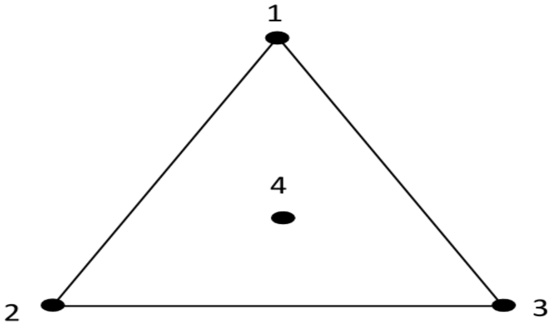


Figure 1.5: Points 1, 2, and 3 will be used to establish the model, and point 4 will be used to validate it

Table 1.1: The matrix of related experiences:

mixture	x_1	x_2	x_3	Cold resistance (°C)
1	1	0	0	-40.5
2	0	1	0	-12.5
3	0	0	1	-19
4	1/3	1/3	1/3	-26.9

The model is written as follows:

$$y = -40.5x_1 - 12.5x_2 - 19.0x_3.$$

The specifics of the cold holding are of ±0.5. The mixture 4 at the domain's center experimental serves to verify the model's validity.

- Amount determined using the model y4 = -24°C.
- Measured value: y4 = 26.9°C.

The difference between the measured value and the calculated value is significantly more than the measure's accuracy. The first-degree model is rejected. The model performs poorly because 26.9 -24 > 0.5°C.

1.5.1.3.2 b) Hypothesis of the Quadratic Model: As we write: :

$$y = a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3$$

The Scheffé network {3,2} is made up of the points 1, 2, 3, 5, 6, and 7 (Figure 1.6)

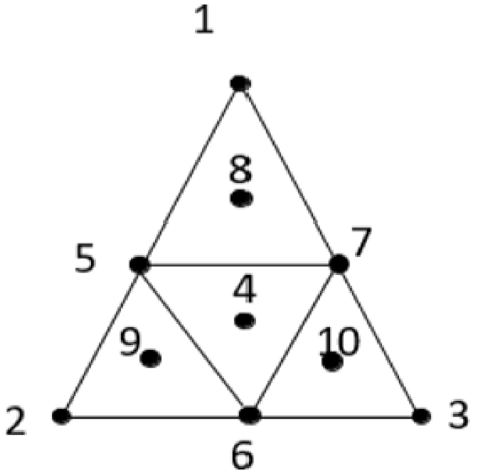


Figure 1.6: Graphical representation of mixtures, points 4, 8, 9 and 10 are used for the validity of the model.

Table 1.2: The tests that need to be performed are shown in the matrix below:

N°	x$_1$	x$_2$	x$_3$	y
1	1	0	0	-40.5
2	0	1	0	-12.5
3	0	0	1	-19
5	1/2	1/2	0	-28.6
6	0	1/2	1/2	-30.8
7	1/2	0	1/2	-18.5

These 6 tests allow you to write a system of 6 equations with 6 unknowns that is very easy to solve. Subject to validity, we therefore have the model:

$$y = -40.5x_1 - 12.5x_2 - 19x_3 - 8.4x_1x_2 + 45x_1x_3 - 60.2x_2x_3$$

We include four more points in the model above to validate it.

Table 1.3: We include four more points in the model above to validate it.

N°	x$_1$	x$_2$	x$_3$	Y measure	Y calculated
4	1/3	1/3	1/3	-26.9	-26.6
8	2/3	1/6	1/6	-29.6	-23.9
9	1/6	2/3	1/6	-24.2	-24.6
10	1/6	1/6	2/3	-23.5	-23.4

Given the strong agreement between the computed and observed values inside the domain, the validity of the quadratic model is established.

1.5.2 Type II mixture design

Type II mixing designs guarantee that $x_i \geq l_i$. Constraints of type $x_i \geq l_i$ are observed quite frequently. For formulation, for example, we determine that a minimum content of i is needed to provide the combination with an improved characteristic worthy of commercial interest.

1.5.2.1 Experimental field

Based on the following relationships, the domain of potential combinations (figure 1.7) is created:

$$\sum_{i}^{K} x_i = 1, \text{ with } : x_1 \geq l_1, x_2 \geq l_2 \text{ and } x_3 \geq l_3$$

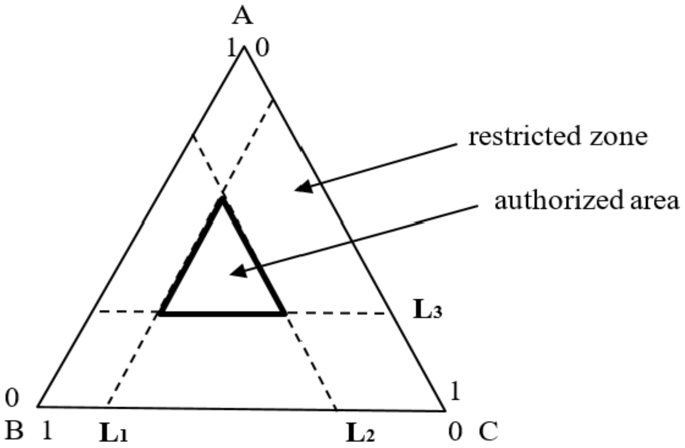


Figure 1.7: The lower limits of all products define two zones: The prohibited zone (shaded zone) and The authorized zone which has the same geometric shape as the initial domain

It is watched that on the off chance that $\sum_i l_i < 1$, the exploratory space decreases to a homothetic figure of the initial space. If not, the space is not dynamic. Moreover, indeed within the nonattendance of an upper restrain, one does exist. Within the case-corresponding figure 1.8, corresponding to the case:

$$x_1 \geq 0.15$$

$$x_2 \geq 0.25$$

$$x_3 \geq 0.10$$

It is obvious that the mixes' concentrations of A and B are separately confined to 0.65, 0.75, and 0.6. The taking after rationale makes calculating these restrictions basic: When each other fixing accomplishes its lower constrain, the constituent i reaches its upper limit U_i.

$$u_i = 1 - \sum_{i \neq j} l_j$$

We find:

$$u_1 = 1 - (0.25 + 0.1) = 0.65$$

$$u_2 = 1 - (0.15 + 0.1) = 0.75$$

$$u_3 = 1 - (0.15 + 0.25) = 0.60$$

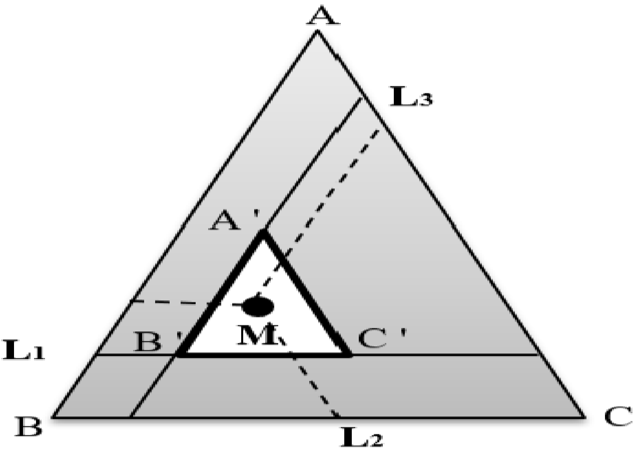


Figure 1.8: Location of the Scheffé network's pseudo-pure bodies.

1.5.2.2 Calculation of the composition of experimental mixtures

The blends to be arranged are components of a reasonable domain-based Scheffé network, whether or not it is centered. Given a Scheffé network with vertices A', B', C',... the associated mixtures are referred to as "pseudo pure bodies" (Figure 1.7). One can represent the composition of a mixture M of the domain in terms of:

- Real pure constituents A, B, C, etc. Let's use $x_1, x_2, ..., x_k$ to indicate the equivalent values. The mixture's true composition is represented by the x_i.
- For pseudo-pure bodies, let the equivalent values be $x'_1, x'_2, \ldots, x'_k$. The model may be calculated with the help of the x'_i.

The connection between the x'_i and the x_i is:

$$x'_i = \frac{x_i - l_i}{1 - \Sigma l_i}$$

1.5.2.3 Example

The three components of a mixture must meet the following requirements: $x_1 \geq 0.4$ and $x_2 \geq 0.3$, where:

$$l_1 = 0.4$$

$$l_2 = 0.3$$

$$l_3 = 0$$

Now let's find the makeup of the combinations shown in A', B', and C' (Figure 1.9):

- In A', we have:

$$x' = 1, x_2 = 0, x_3 = 0$$

From where:

$$1 = \frac{x_1 - 0.4}{1 - 0.7} \Rightarrow x_1 = 0.7$$

$$0 = \frac{x_2 - 0.3}{1 - 0.7} \Rightarrow x_2 = 0.3$$

$$0 = \frac{x_3 - 0}{1 - 0.7} \Rightarrow x_3 = 0$$

- In B', we have:

$$x'_1 = 0, x'_2 = 1, x'_3 = 0$$

From where

$$0 = \frac{x_1 - 0.4}{1 - 0.7} \Rightarrow x_1 = 0.4$$

$$1 = \frac{x_2 - 0.3}{1 - 0.7} \Rightarrow x_2 = 0.6$$

$$0 = \frac{x_3 - 0}{1 - 0.7} \Rightarrow x_3 = 0$$

- In C', we have:

$$x'_1 = 0, x'_2 = 0, x'_3 = 1$$

$$0 = \frac{x_1 - 0.4}{1 - 0.7} \Rightarrow x_1 = 0.4$$

$$1 = \frac{x_2 - 0.3}{1 - 0.7} \Rightarrow x_2 = 0.3$$

$$0 = \frac{x_3 - 0}{1 - 0.7} \Rightarrow x_3 = 0.3$$

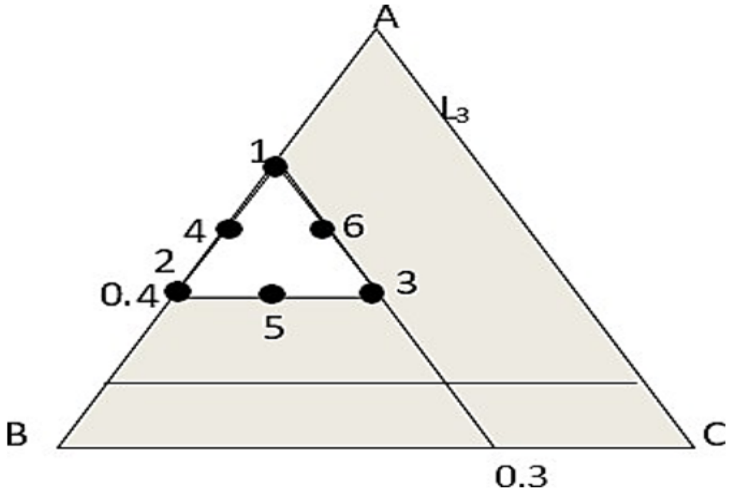


Figure 1.9: Position of the pseudo bodies and other mixtures of the Scheffé network.

The linear model's coefficients will be found at points 1, 2, and 3, and its validity will be examined at point 7:

Table 1.4: Experimental plan to establish the first-degree model and test its validity

Points	A'	B'	C'	Module Elasticity
1	1	0	0	14150
2	0	1	0	17550
3	0	0	1	6450
7	1/3	1/3	1/3	10850

Points 1,2 and 3 give:

$$a_1' = 14150$$

$$a_2' = 17550$$

$$a_3' = \ 6450$$

The model to be tested is written:

$$y = 14150x_1' + 17550x_2' + 6450x_3'$$

At point 7, the predicted value is $y = 12717$, which is 1867 points different from the observed result. We reject the first-degree model. We finish with the centers of gravity, 8, 9, and 10, and the midpoints of the edges, points 4, 5, and 6, just as in the case of the type I mixing design (Table 1.5).

Table 1.5: Mixtures for testing the quadratic model

Points	A'	B'	C'	Module Elasticity
1	1	0	0	14150
2	0	1	0	17550
3	0	0	1	6450
4	0.5	0.5	0	15550
5	0	0.5	0.5	10400
6	0.5	0	0.5	8600
7	1/3	1/3	1/3	10850
8	0.667	0.167	0.167	12100
9	0.167	0.667	0.167	14250
10	0.167	0.167	0.667	8300

The six model coefficients may be obtained by completing the first six bridges. We find:

$$a'1 = 14150 \qquad a'12 = -1200$$

$$a'2 = 17550 \qquad a'13 = -6800$$

$$a'3 = 6450 \qquad a'23 = -6400$$

The quadratic model expressed as a function of pseudo pure bodies has the equation:

$$y = 14150x'_1 + 17550x'_2 + 6450x'_3 - 1200x'_1x'_2 - 6800x'_1x'_3 - 6400x'_2x'_3$$

Points 7 to 10 are used to test the validity of the model, which translates into:

Table 1.6: Experimental plan to establish the first-degree model and test its validity

Points	Elasticity module	Calculated Module	Gap
7	10850	11117	-267
8	12100	12367	-267
9	14250	14100	150
10	8300	8083	217

Any deviation is permissible. It is valid to use the quadratic model. It is best to write the model in terms of the genuine compositions x_i when employing it. We get:

$$y = 13623x_1 + 27027x_2 + 36343x_3 - 16914x_1x_2 - 78732x_1x_3 - 71762x_2x_3$$

1.5.3 Type III mixture design

Plans classified as type III are those in which $l_i \leq x_i \leq u_i$, where l_i and u_i stand for the constituent it's lower and upper constraint, respectively. These are the most prevalent by far. They have greater complexity as well.

1.5.3.1 The Principle approach

You need to have $x_i \geq l_i$ to feel a significant effect of the constituent i on the properties of the mixture, but you also need to have $x_i \leq u_i$ for cost reasons for example (Figure 1.10).

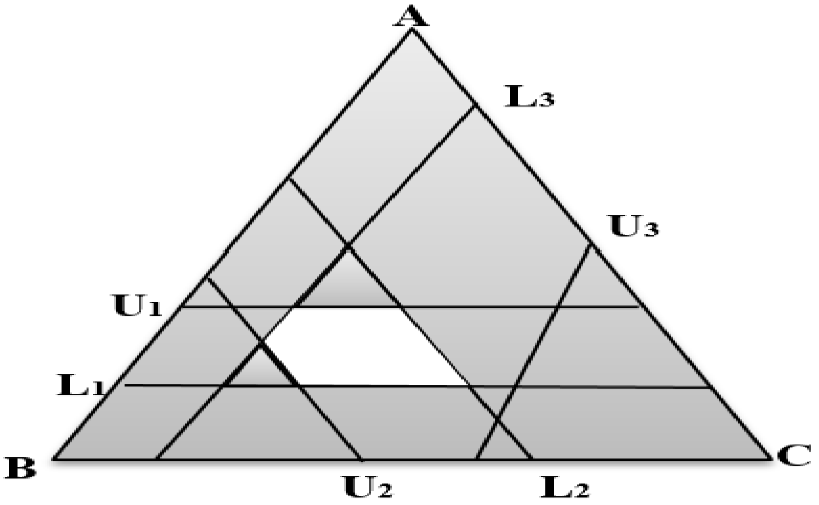


Figure 1.10: Domain of possible mixtures when there is a lower limit and an upper limit.

Adding double restrictions usually results in the experimental domain taking on a different structure. It is evident that the domain's form and vertex count are contingent upon the restrictions l_i and u_i. Scheffé networks are outdated and unusable. The next steps will be as follows:

- Find the domain's vertices, edges, faces, and so on; keep in mind that the domain is always convex in shape.
- Form a set C of points that are probably going to be in the plan. These potential locations are the vertices, the midpoints of the edges, the centers of the faces, etc. by comparison with what we have seen thus far. The decision is based on the desired level of the model.
- Choose the N plan points from the C candidates using the maximal determinant criterion or an equivalent criterion.

1.5.3.2 Actual boundaries of the domain

The limits l_i and u_i being fixed for each of the constituents, we must first check the validity of the whole. Let's ask:

$$L = \sum_{i=1} l_i \text{ and } U = \sum_{i=1} u_i$$

The domain can only exist if $L < 1$ and $U > 1$. Furthermore, we must see in Figure 1.10 that upper limits may never be reached. To find the real upper limit of a constituent i, the upper limit cannot exceed the quantity: $1 - \sum_{i \neq j} l_i$. Subtracting l_i from both sides of the inequality:

$$u_i \leq 1 - \sum_{i \neq j} l_i$$

We obtain :

$$u_i - l_i \leq 1 - L$$

or:

$$R_i \leq 1 - L$$

A constituent's limit needs to be adjusted if its concentration range R_i is greater than the number $1 - L$.

It is simple to determine the domain's vertices and their coordinates geometrically for mixtures consisting of three ingredients. An algorithm is required beyond that. In the following example, we explain how to apply the Snee and Marquardt [Snee and Marquardt (1974)] algorithm to formulate a lubricant using four bases. The following table provides a summary of their respective boundaries and domains:

Table 1.7: Experimental plan to establish the first-degree model and test its validity

Constituent	l_i	U_i	R_i
1	0.25	0.45	0.20
2	0.00	0.20	0.20
3	0.20	0.45	0.25
4	0.00	0.15	0.15

Existence of the experimental domain: Let's calculate the sum of the lower limit $L = \sum_{i=j} l_i$ and higher $U = \sum_{i=j} u_i$. For that the experimental domain exists, we must have $L < 1$ and $U > 1$

$$L = 0.25 + 0 + 0.20 + 0 = 0.45 < 1$$

$$U = 0.45 + 0.20 + 0.45 + 0.15 = 1.25 > 1$$

The experimental domain exists.

1.5.3.3 Consistency of explicit constraints. cohérent domain

To do this, we will compare the ranges $R_i = u_i - l_i$ each constituent to

$$R_l = 1 - L = 1 - \sum_{i=1}^{3} l_i$$

(linear measurement of simplex A) and

$$R_u = U - 1 = \sum_{i \neq 1} u_i - 1$$

The lower and upper constraints or limits are denoted respectively by l_i and u_i. Ranges of constituents (or areas of constituent concentrations):

$$R_1 = 0.45 - 0.25 = 0.20$$

$$R_2 = 0.20 - 0.00 = 0.20$$

$$R_3 = 0.45 - 0.20 = 0.25$$

$$R_4 = 0.15 - 0.00 = 0.15$$

There is equality in the linear measurements of simplexes A and B.

$$R_l = 1 - L = 1 - 0.45 = 0.55$$

$$R_u = U - 1 = 1.25 - 1 = 0.25$$

The top limit of the associated ingredient is incompatible if the extent R_i is more than the number R_l. This higher restriction, known as the implicit constraint, needs to be changed with the new one, $u_i = l_i + R_l$. Since all of the R_i in our situation are smaller than R_l, there is no need to adjust the upper limit. As a result, there is no need to adjust the boundaries because the suggested experimental domain is coherent.

1.5.3.4 Algorithme of Snee and Marquardt

- Step 1: We classify the constituents in ascending order of their range of variation

R_4	$< R_1 =$	$R_2 <$	R_3
0.15	0.20	0.20	0.45

- Step 2:: We construct a complete factorial matrix $2^{k-1} = 2^3$ with the first $(k-1)$ constituting x_4, x_1 and x_2, the lower constraint l_i corresponds to level -1 and the upper constraint u_i at level +1. We fill the column of the last constituting so as to respect the general constraint of mixtures $\sum_{i=1}^{4} x_i = 1$: Application to our example leads to the following table:

Table 1.8: We include four more points in the model above to validate it.

Constituent	x_4	x_1	x_2	x_3	$0.20 < x_3 < 0.45$
1	0.00	0.25	0.00	0.75	Outside the domain
2	0.15	0.25	0.00	0.60	Outside the domain
3	0.00	0.45	0.00	0.55	Outside the domain
4	0.15	0.45	0.00	0.40	
5	0.00	0.25	0.20	0.55	Outside the domain
6	0.15	0.25	0.20	0.40	
7	0.00	0.45	0.20	0.35	
8	0.15	0.45	0.20	0.20	

- Step 3: Check if the concentrations of the last constituent satisfy the imposed constraints. If yes, the corresponding line is a vertex of the domain. If not, the last constituent is brought back to its closest limit. It is then necessary to readjust the sum of the concentrations to 1 by modifying one of the concentrations of the first $k-1$ constituents. All possible solutions correspond to new highs.

Line 1

On the first line of the table, the concentration x_3 is reduced from 0.75 to 0.45. So 0.30 is to be added either to x_4, x_1, or x_2. No distribution can be accepted.

Line 2

The concentration x3 drops to 0.45 from 0.60. It is therefore necessary to add 0.15 to x_4, x_1, or x_2. We acquire the pair of vertices:

$$x_4 = 0.15. \qquad x_1 = 0.40, \qquad x_2 = 0.00, \qquad x_3 = 0.45,$$

$$x_4 = 0.15. \qquad x_1 = 0.25, \qquad x_2 = 0.15, \qquad x_3 = 0.45,$$

Line 3

The concentration x3 is now just 0.45 instead of 0.55. It is therefore necessary to add 0.10 to x_4, x_1, or x_2. We acquire the pair of vertices:

$$x_4 = 0.10. \qquad x_1 = 0.45, \qquad x_2 = 0.00, \qquad x_3 = 0.45,$$

$$x_4 = 0.00. \qquad x_1 = 0.45, \qquad x_2 = 0.10, \qquad x_3 = 0.45,$$

Line 4

There is a decrease in the x3 concentration from 0.55 to 0.45. It is therefore necessary to add 0.10 to x_4, x_1, or x_2. We acquire the pair of vertices:

$$x_4 = 0.10. \qquad x_1 = 0.25, \qquad x_2 = 0.20, \qquad x_3 = 0.45,$$

$$x_4 = 0.00. \qquad x_1 = 0.25, \qquad x_2 = 0.20, \qquad x_3 = 0.45,$$

The following table compiles the coordinates of the polyhedron's ten vertices. The measurements of the octane number, which will be utilized to create the first-degree model, are given in the last column.

Table 1.9: Vertices of the experimental domain and their composition.

Summit	Aromatics A x_1	Olefins B x_2	Paraffins C x_3	Oxygen D x_4	Octane Number measured y
1	0.40	0	0.45	0.15	102.4
2	0.25	0.15	0.45	0.15	97.7
3	0.45	0	0.45	0.10	102.8
4	0.45	0.10	0.45	0	100.6
5	0.45	0	0.40	0.15	103.1
6	0.25	0.20	0.45	0.10	96.8
7	0.35	0.20	0.45	0	98.0
8	0.25	0.20	0.40	0.15	97.4
9	0.45	0.20	0.35	0	99.6
10	0.45	0.20	0.20	0.15	100.3

- Step 4: Search for the edges, their length and the composition of their center. After the vertices, we must identify the edges, faces, etc., hyper faces and other boundaries of the experimental domain. Depending on the number k of constituents, the domain is limited by:

- Edges ($k = 3$)
- Edges and faces ($k = 4$)
- Edges, faces, etc., hyper faces of dimension $r = k - 2$

All mixtures of a boundary of dimension r have in common the concentrations of p constituents, p given by the relation: $p = k - r - 1$. Thus, mixtures located on the same edge ($r = 1$)have : $p = 4 - 1 - 1 = 2$. The search for the boundaries of the domain is carried out from the summit table. Starting from vertex1, we search the remainder of the table for vertices which have p common values. We repeat the process with vertex 2. Only concentrations l_i are taken into account Let's apply the approach to our example.

Edges including vertex 1

Vertex 1 has the composition: $x_1 = 0.40$, $x_2 = 0.00$,$x_3 = 0.45$, $x_4 = 0.15$. Let's find the values: $x_3 = 0.45$ and $x_4 = 0.15$ on line 2 of the table:

$$\begin{array}{llll} x_2 = 0 & \text{et} & x_3 = 0.45, & \text{sur la ligne 3} \\ x_2 = 0 & \text{et} & x_3 = 0.15, & \text{sur la ligne 5} \end{array}$$

We identified 3 edges of the domain, edges 1-2,1-3 and 1-5. The middle of the ridge 1-2 has the composition:

$$\begin{aligned} x_1 &= \frac{0.40 + 0.25}{2} = \frac{0.65}{2} = 0.325, \\ x_2 &= \frac{0 + 0.15}{2} = \frac{0.15}{2} = 0.075, \\ x_3 &= \frac{0.45 + 0.45}{2} = \frac{0.90}{2} = 0.45, \\ x_4 &= \frac{0.15 + 0.15}{2} = \frac{0.30}{2} = 0.15. \end{aligned}$$

The compositions of the midpoints of edges 1–3 and 1–5 are obtained in the same way and have respectively the values:

$$\begin{array}{llll} x_1 = 0.425. & x_2 = 0, & x_3 = 0.425, & x_4 = 0.125, \\ x_1 = 0.425. & x_2 = 0, & x_3 = 0.425, & x_4 = 0.15, \end{array}$$

Edges starting from vertex 2

Vertex 2 has the composition: $x_1 = 0.25$, $x_2 = 0.15$, $x_3 = 0.45$, $x_4 = 0.15$. We find the values:$x_1 = 0.25$, $x_3 = 0.45$ on line 6 and the values $x_1 = 0.25$, $x_4 = 0.15$ on line 8. We have identified the two edges 2 - 6 and 2 - 8 whose center compositions are respectively:

$$\begin{array}{llll} x_1 = 0.25. & x_2 = 0.175, & x_3 = 0.45, & x_4 = 0.125, \\ x_1 = 0.25. & x_2 = 0.175, & x_3 = 0.425, & x_4 = 0.15, \end{array}$$

Edges starting from vertex 3

Vertex 3 has the composition:$x_1 = 0.45$, $x_2 = 0$, $x_3 = 0.45$, $x_4 = 10$. We find the values:$x_1 = 0.45$, $x_3 = 0.45$ on line 4 and the values $x_1 = 0.45$, $x_2 = 0$ on line 5. We have identified the edges

3- 4 and 3-5 starting from vertex 3 and whose midpoints have respectively the composition:

$$x_1 = 0.45. \qquad x_2 = 0.05, \qquad x_3 = 0.45, \qquad x_4 = 0.05,$$

$$x_1 = 0.45. \qquad x_2 = 0, \qquad x_3 = 0.425, \qquad x_4 = 0.125,$$

Edges starting from vertex 4

Vertex 4 has the composition: $x_1 = 0.45$, $x_2 = 0.1$, $x_3 = 0.45$,$x_4 = 0$.We find the value $x_3 = 0.45$, $x_4 = 0$ on line 7 and the values $x_1 = 0.45$, $x_4 = 0$ on line 9. We have identified edges 4-7 and 4-9 whose midpoints have respectively the composition:

$$x_1 = 0.40, \qquad x_2 = 0.15, \qquad x_3 = 0.45, \qquad x_4 = 0$$

$$x_1 = 0.45. \qquad x_2 = 0.15, \qquad x_3 = 0.40, \qquad x_4 = 0.$$

Edges starting from vertex 5

Summit 5 has the following composition:

$$x_1 = 0.45. \qquad x_2 = 0, \qquad x_3 = 0.40, \qquad x_4 = 0.15$$

We find the value: $x_1 = 0.45$, $x_4 = 0.15$ on line 10 We have identified the edge 5 -10 whose middle has the composition:

$$x_1 = 0.45. \qquad x_2 = 0.10, \qquad x_3 = 0.30, \qquad x_4 = 0.15$$

Edges starting from vertex 6

Vertex 6 has the composition: $x_1 = 0.25$, $x_2 = 0.20$, $x_3 = 0.45$, $x_4 = 0.10$. We find the values $x_2 = 0.20$, $x_3 = 0.45$ on line 7 and the values $x_1 = 0.25$, $x_2 = 0.20$ on line 8. We have detected the edges $6 - 7$ and $6 - 8$ whose backgrounds have respectively the composition:

$$x_1 = 0.30, \qquad x_2 = 0.20, \qquad x_3 = 0.45, \qquad x_4 = 0.05$$

$$x_1 = 0.25. \qquad x_2 = 0.20, \qquad x_3 = 0.425, \qquad x_4 = 0.125$$

Edges starting from vertex 7

Vertex 7 has the composition: $x_1 = 0.35$, $x_2 = 0.20$, $x_3 = 0.45$ $x_4 = 0$. We find the values$x_2 = 0.20$, $x_4 = 0$ on line 9. We have just detected the edge $7 - 9$ whose composition of its middle is:

$$x_1 = 0.40, \qquad x_2 = 0.20, \qquad x_3 = 0.40, \qquad x_4 = 0$$

Edges starting from vertex 8

Vertex 8 has the composition: $x_1 = 0.25$, $x_2 = 0.20$; $x_3 = 0.40$, $x_4 = 0.15$. We find the values $x_2 = 0.20$, $x_4 = 0.15$ on line 10. We have just detected the edge 8 –10 whose middle composition is:

$$x_1 = 0.35, \qquad x_2 = 0.20, \qquad x_3 = 0.30, \qquad x_4 = 0.15$$

Edges starting from vertex 9

Vertex 9 has the composition: $x_1 = 0.45$, $x_2 = 0.20$, $x_3 = 0.35$ $x_4 = 0$. We find the values $x_1 = 0.45$, $x_2 = 0.20$ on line 10. We have just detected the edge 9 -10 whose middle composition is:

$$x_1 = 0.45, \qquad x_2 = 0.20, \qquad x_3 = 0.275, \qquad x_4 = 0.075$$

In total the domain has 15 edges listed in the following table:

Table 1.10: We include four more points in the model above to validate it.

Edges	x$_1$	x$_2$	x$_3$	x$_4$
1– 2	0.325	0.075	0.45	0.15
1– 3	0.425	0	0.45	0.125
1– 5	0.425	0	0.425	0.15
2 – 6	0.25	0.175	0.45	0.125
2 – 8	0.25	0.175	0.425	0.15
3 – 4	0.45	0.05	0.45	0.05
3 – 5	0.45	0	0.425	0.125
4 – 7	0.40	0.15	0.45	0
4 – 9	0.45	0.15	0.40	0
5 –10	0.45	0.10	0.30	0.15
6 – 7	0.30	0.20	0.45	0.05
6 – 8	0.25	0.20	0.425	0.125
7 – 9	0.40	0.20	0.40	0
8 –10	0.35	0.20	0.30	0.15
9 –10	0.45	0.20	0.275	0.075

- Step 5: Search for faces and the composition of their center. The faces have the dimension $r = 2$ The faces which constitute the boundaries of the domain therefore have in common the concentrations of $p = k - r - 1 = 1$ constituents. The search for faces is always done from the vertex composition table.

Faces including vertex1:

Vertex 1 has the composition: $x_1 = 0.40$, $x_2 = 0$, $x_3 = 0.45$, $x_4 = 0.15$, vertices 3 and 5 define with vertex 1 a face characterized by the common concentration $x_2 = 0$ of constituent B and whose center has the composition:

$$x_1 = \frac{0.40 + 0.45 + 0.40}{3} = \frac{1.25}{3} = 0.4167,$$
$$x_3 = \frac{0.45 + 0.45 + 0.40}{3} = \frac{1.30}{3} = 0.4333,$$
$$x_4 = \frac{0.15 + 0.10 + 0.15}{3} = \frac{0.40}{3} = 0.1333.$$

Likewise, face 1– 2 – 3 – 4 – 6 – 7 is characterized by the common concentration $x_3 = 0.45$ of constituent C and face 1– 2 – 5 – 8 –10 by $x_4 = 0.15$ of constituent D. The centers respectively have the value:

$x_1 = 0.3583,$ $x_2 = 0.1083,$ $x_3 = 0.45,$ $x_4 = 0.0833$

$x_1 = 0.36,$ $x_2 = 0.11,$ $x_3 = 0.38,$ $x_4 = 0.15$

Faces including vertex 2 ,3, 4, 6 Using the same procedure, we will successively identify the faces 2 – 6 – 8, 3 – 4 – 5 – 9 –10, 4 – 7 – 9 and 6 – 7 – 8 – 9 –10 whose characterization and the composition of the centers are reported in the following table. We get a total of 7 faces:

Table 1.11: Faces of the experimental domain and composition of their center.

Face	Center of faces	Characterization
1– 3 – 5	0.4333 0 0.4333 0.1333	B = 0
1– 2 – 3 – 4 – 6 – 7	0.3583 0.1083 0.45 0.0833	C = 0.45
1– 2 – 5 – 8 –10	0.36 0.11 0.38 0.15	D = 0.15
2 – 6 – 8	0.25 0.1833 0.4333 0.1333	A = 0.25
3 – 4 – 5 – 9 –10	0.45 0.10 0.37 0.08	A = 0.45
4 – 7 – 9	0.4167 0.1667 0.4167 0	D = 0
6 – 7 – 8 – 9 –10	0.35 0.20 0.37 0.08	B = 0.20

- Step 6: Composition of the center of gravity G

$x_1 = 0.375 x_2 = 0.125$ $x_3 = 0.405$ $x_4 = 0.095$

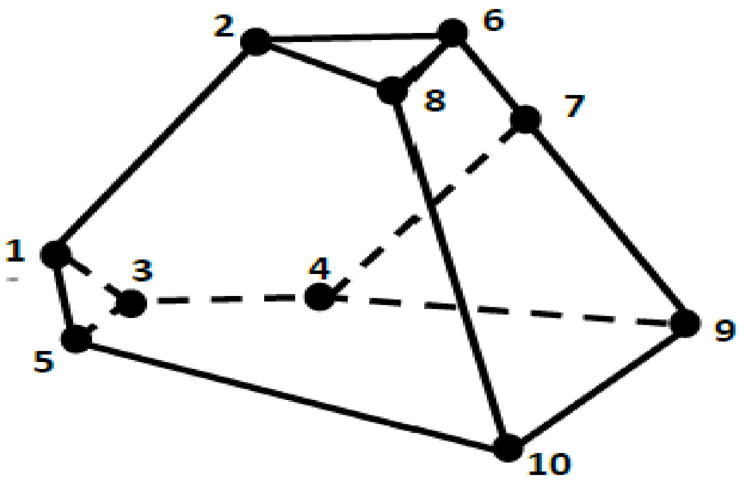


Figure 1.11: Domain of possible mixtures when there is a lower limit l and an upper limit u.

Chapter 2

Optimality criterion D

In the field of experimental design, the D-optimality criterion plays a crucial role in the design and evaluation of experimental plans. This criterion, also known as the D-optimality criterion, is used to assess the quality of experimental plans in terms of precision and efficiency. Essentially, it aims to minimize the variance of estimates of the parameters of the underlying statistical model, thereby obtaining more precise and reliable estimates.

In this chapter, we will delve deep into the D-optimality criterion and its significance in the design of experimental plans. We will examine its fundamental principles, practical applications, and implications for experimental research. Additionally, we will discuss the advantages and limitations associated with the use of this criterion, as well as commonly used methods for optimizing experimental plans based on this criterion.

2.1 Basic Concepts of Optimality

Objective Function: In mixture design, the objective function defines the target for optimization, whether it involves maximizing the response variable or minimizing variability. It typically mirrors the desired properties or traits of the mixture under scrutiny.

- **Constraints:** Constraints in mixture design delineate the restrictions or limits on the proportions of components within the mixture. These constraints often stipulate upper and lower bounds on component proportions and ensure that the proportions sum to unity. They are vital for ensuring the practicality and feasibility of the resulting mixture.

- **Optimal Mixture:** The optimal mixture represents the blend of component proportions that satisfies the objective function while adhering to the specified constraints. It epitomizes the most desirable mixture composition in terms of achieving the desired response variable or properties.

- **Local Optima vs. Global Optima:** Similar to other optimization challenges, mixture design may yield multiple optimal solutions. A local optimum denotes a solution that is optimal within a specific region of the solution space but may not be globally optimal. In contrast, a global optimum signifies the best solution across the entire solution space, embodying the overall optimal mixture composition.

- **Optimization Algorithms:** Optimization algorithms are computational methodologies employed to explore and identify optimal mixture compositions. These algorithms navigate the solution space of component proportions, evaluating potential mixtures based on the objective function and constraints. Common optimization algorithms utilized in mixture design include simplex optimization, gradient-based approaches, and evolutionary algorithms.

- **Sensitivity Analysis:** Sensitivity analysis in mixture design assesses the robustness of the optimal mixture to variations in input parameters or constraints. It aids in pinpointing critical factors influencing the optimality of the mixture composition and provides insights into the stability and reliability of the optimal solution.

In summary, achieving optimality in mixture design involves identifying the best combination of component proportions to fulfill the desired properties or characteristics of the mixture while considering imposed constraints and trade-offs. By carefully balancing these factors, optimal mixture compositions can be determined to meet specific objectives across various applications, including product formulation, process optimization, and experimental design.

2.2 The D-Optimal Approach

A D-optimal design is a computer-aided design that incorporates the best subset of all possible experiments. A predetermined number of design runs and a predetermined criterion are used in a selection process to determine which design is the best.

2.2.1 Candidate Set

All theoretically and practically feasible experiments are contained in a matrix known as the candidate set, in which "each row represents an experiment and each column a variable" [de Aguiar et al. (1995)] . This so-called matrix of candidate points is represented by ξ_N and consists of N rows.

The candidate set for a straightforward investigation with two factors 1 and 2 has two columns and four rows. Since we only take into account the two experiments where the factors have a minimum or maximum value, we obtain four rows. Although it isn't taken into account in this fundamental example, choosing different experiments can be helpful in certain situations. The candidate set is displayed in the extended notation in equation 2.3.

$$\xi_4 = \begin{bmatrix} -1 & -1 \\ -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{2.1}$$

2.2.2 Design Matrix

An $n\ x_p$ matrix with p coefficients is the basis of the design matrix X. The experimenter can select the number of rows n, which indicates how many experiments are included in the design. It is simple to construct the design matrix given a model and a candidate matrix. Based on the terms in the model, each column contains a combination of the candidate set factors. The matrix is sometimes referred to as the model matrix, but most of the time, the term refers to a $n\ x_p$ matrix that has the model-dependent rows for each candidate [de Aguiar et al. (1995)] .

For a basic example with $n = 4$ design runs, we utilize the model from equation 2.4 and the previously mentioned candidate set .

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon \tag{2.2}$$

As a result, all of the candidates from are used in the design of the model matrix, which has four rows and four columns. Typically, only a small portion of the model matrix is present in the candidate set, which has far more experiments.

$$X = \begin{bmatrix} 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \tag{2.3}$$

There are only ones in the first column of X because it represents the constant term β_0. The model terms (2,1) and (2,2) , which are drawn from the candidate set ξ_4, represent the factors under investigation in columns two and three. An interaction between the two factors is represented by the final column of X. Therefore, we must multiply the two candidate set columns. A larger candidate set expands the pool of potential EN subsets, necessitating the use of specific criteria in the design matrix selection process. According to [de Aguiar et al. (1995)]

, the optimal combination of these points is referred to as optimal, and the corresponding design matrix is known as the optimal design matrix ${}^tX$. In section 2.3, we address the various criteria.

2.2.3 Information and Dispersion Matrix

In order to select the optimal design using the criteria that will be discussed later, two additional types of matrices must be defined. The so-called information matrix ${}^tXX$ is the first one. The design matrix X' and X itself are multiplied by their transpose to create this matrix. The inverse matrix of this computation is the dispersion matrix $({}^tXX)^{-1}$ [de Aguiar et al. (1995)]. The least-squares estimate for an assumed model provides the context for these equations. An example using matrix notation:

$$y = \mathbf{X}\beta + \varepsilon$$

has the optimal collection of coefficients based on the least squares provided by :

$$\hat{\beta} = (X'X)^{-1}X'y. \tag{2.4}$$

[Box et al. (1978)] and [Wu and Hamada (2000)] provide more details on the least square estimator.

2.3 Key Features of Optimal Designs

- **Efficiency:** Optimal designs excel in resource utilization, demanding a minimal number of experimental runs to achieve the desired precision in parameter estimation.

- **Precision:** Optimal designs provide accurate estimates of model parameters by strategically allocating experimental runs to areas within the experimental space where information is most pertinent.

- **Robustness:** Optimal designs exhibit resilience to variations in underlying model assumptions, ensuring dependable estimates even in the presence of noise or inaccuracies in the model.

- **Balance:** Optimal designs often strike a delicate balance between exploring the entire experimental space and focusing on specific areas of interest, ensuring comprehensive coverage while maximizing information retrieval.

- **Criterion Optimization:** Optimal designs are developed based on specific optimization criteria, such as D-optimality, A-optimality, or E-optimality, which assess the quality of the design in terms of precision, efficiency, or other pertinent factors.

2.4 D-optimal design

D-optimal design is a concept in experimental design that is used to select the most efficient design for an experiment, particularly when classical symmetrical designs are not suitable. It is one of several criteria that can be used to determine the optimality of a design.

In conclusion, while designing experiments, mixed designs provide a distinctive test area that has to be taken into account. In contrast to more conventional design possibilities, they concentrate on a blending of components that cannot be changed separately, producing a limited design zone.

2.4.1 Principle of D-optimal design

Selecting the most effective selection of experimental points to elicit as much information as possible from a study is at the heart of the D-optimal design principle. It accomplishes this by optimizing parameter estimate precision inside a pre-established model. Here is a summary of the salient features:

- **Maximizing Information Matrix Determinant:** The goal of D-optimal design is to maximize the information matrix's determinant ($|{}^tXX|$). The link between the desired estimate of the model parameters and the design points (X) is captured in this matrix. Smaller variances for the estimated parameters are indicative of a more informative design, as indicated by a higher determinant.

- **Minimizing Generalized Variance:** To put it another way, the goal of D-optimal design is to reduce the total amount of uncertainty surrounding the model's predicted parameters. It guarantees greater coverage of the experimental region and lowers the generalized variance of the estimates by strategically dispersing the experimental locations.

- **Model-Dependent:** D-optimal designs, in contrast to certain classical designs, are customized for a particular model. Before the design can be created, the experimenter must specify the model they wish to fit the data to. This makes it possible for the algorithm to select locations that are most useful for determining the parameters in that specific model.

- **Here's an analogy:** picture yourself thoroughly mapping out a brand-new landscape. D-optimal design might be compared to carefully selecting observation sites to provide the best possible image of the topography's features. Instead of concentrating on a specific location, you can make a map that is more precise by extending across valleys, peaks, and slopes.

- **Additional Points:** Computer algorithms are frequently used to create D-optimal designs. For a given model, these algorithms look for the set of points that maximizes the determinant of the information matrix.While D-optimal designs are highly precise, they may not always be orthogonal (meaning factors are independent). This can lead to some correlation between the estimated effects.

2.4.2 The need for D-optimal designs

2.4.2.1 Irregular Experimental Regions

The variables under investigation define the experimental region. The area's form is determined by the quantity, kind, and overall arrangement of all the components. The best method for illustrating the area is to use a straightforward plot. [Eriksson et al. (2000)].

There are no limitations on the problem formulation in any of these domains. A constraint would be something like not being able to conduct experiments in a certain area of the region. An illustration of a quadratic design with a constraint in the upper right corner may be found in Figure 2.1's middle column. Since this corner is not examined, a standard design is not appropriate. This may be because the experimenter wishes to avoid certain factor combinations or because external influences hinder further investigation of this corner [Eriksson et al. (2000)]. The uneven experimental region shown in Figure 2.1 can be handled in two different ways. The simplest would be to reduce the area till it possesses a quadratic form once more, but this would cause distortion to the entire study and is not advised. The development of a computer-aided D-optimal design is a more efficient method. As demonstrated by the bottom right square of the four-in

In Figure 2.1, the D-optimal algorithm chooses two points on the border of the constraint instead of the excluded corner. This increases the number of design runs but is essential to deal with the complexity of the constricted experimental region. In addition to this, the center-point is manipulated [Eriksson et al. (2000)].

An erratic experimental region can also be a feature of mixture design. A design containing the three factors A, B, and C is depicted by the triangle in Figure 2.1. All of the triangle's corners are normally attainable, however in this case, the factors are limited. A limitation for mixing factors is when the lower and upper bounds deviate from 0 and 1 [Eriksson et al. (2000)].

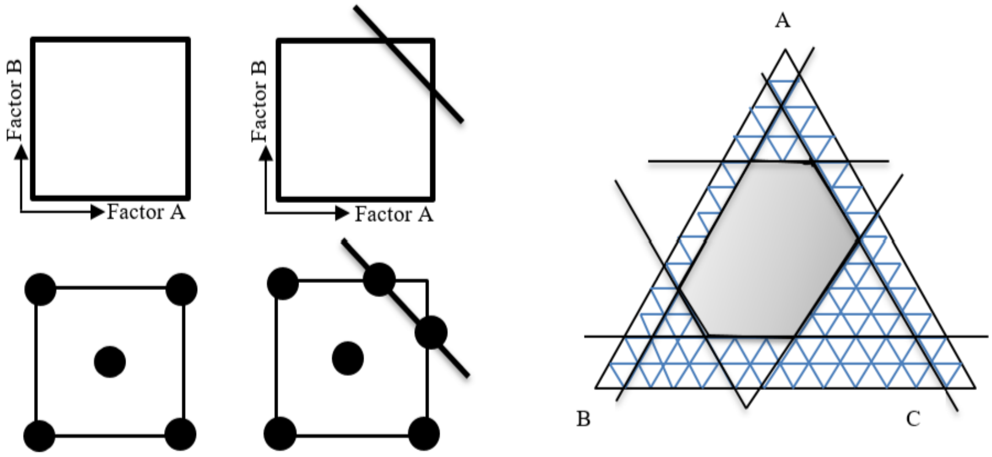


Figure 2.1: Examples for Irregular Experimental Regions:

2.4.2.2 Inclusion of Already Performed Experiments

Sometimes an experimenter wishes to include the results of a particular number of trials he has already conducted into his ongoing research. The experiments cannot be included into traditional designs. These additional runs can be included into the design and taken into consideration during the creation process by using D-optimal designs. [Eriksson et al. (2000)].

2.4.2.3 The use of qualitative factors

A qualitative factor lacks a continuous scale and only has discrete values. The number of runs for a typical design sharply increases if the number of these discrete phases exceeds two. An investigation with two qualitative and one quantitative element is depicted in Figure 2.2. To solve this problem using three and four discrete values for each of the two qualitative elements, a full factorial design would require 4*3*2=24 design runs.

Only twelve experiments remain in the design runs when using the D-optimal technique. To ensure a balanced design throughout the entire experimental zone, these experiments—represented as filled circles in figure 2.2—were selected. Every level of a qualitative element in a balanced design has an equal number of runs [Eriksson et al. (2000)].

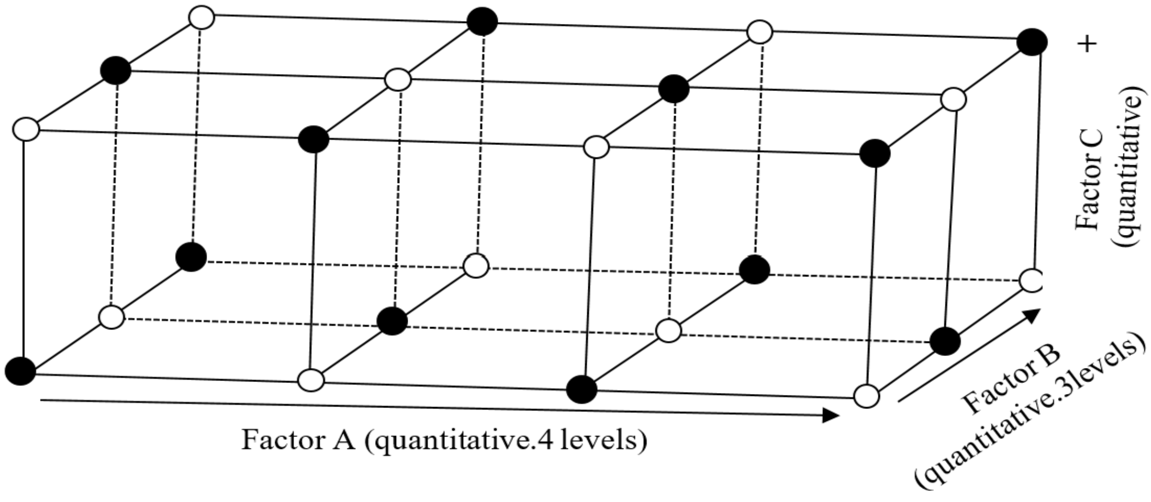

+
Factor C (quantitative)
Factor B (quantitative.3levels)
Factor A (quantitative.4 levels)

Figure 2.2: Design With Multi-Level Qualitative Factors

2.4.2.4 Reducing the number of experiments

The more elements there are in the classical designs, the less efficient they are. Examples of the minimum number of runs for a D-optimal design, a fractional factorial, and a complete factorial are shown in Table 2.1. When compared to classical designs with an increasing number of components, the required runs for a D-optimal design are always less and do not increase as quickly [Umetrics (2006)].

Table 2.1: Experimental plan to establish the first-degree model and test its validity

Factors	Full Factorial	Fractional Factorial	D-Optimal
5	32	16	16
6	64	32	28
7	128	64	35
8	256	64	43
9	512	128	52

2.4.2.5 Fitting of special regression models

D-optimal designs allow for many modifications to be made to the underlying model. Equation 2.5 demonstrates that if the experimenter is aware that a certain term is not crucial to the answer, they can eliminate it. This permits lowering the number of runs without having a major affect on the inquiry.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{12} x_1 x_2 + \beta_{13} x_1 x_3 + \beta_{23} x_2 x_3 + \epsilon \tag{2.5}$$

The inclusion of a single higher-order term is the second potential model alteration. When using classical designs, one can only make complete model changes, such as switching from an interaction to a quadratic model. On the other hand, independent model terms can be added to D-optimal designs. An illustration of a linear model with an extra interaction term is provided by the following equation.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_{23} x_{23} + \epsilon \tag{2.6}$$

2.4.3 Example of using a D-optimal design

This example, which comes from [Eriksson et al. (2000)] , employs a geometric demonstration of the D-optimal technique using an investigation involving two elements, x_1 and x_2 Three levels of analysis are performed on the two factors: -1, 0 and 1. Table 2.2 lists the corresponding candidate set.

Table 2.2: Candidate Set for Two Factors with Three Levels.

N_r	1	2	3	4	5	6	7	8	9
x_1	-1	-1	-1	0	0	0	1	1	1
x_2	-1	0	1	-1	0	1	-1	0	1

If the candidate set were shown using matrix notation, the matrix ξ_9 would have $N = 9$ experiments and two columns for each factor. Plotting the nine choices from ξ_9 over the experimental region is shown in Figure 2.3. Out of this candidate set, we have $9!/(3!6!) = 84$ potential subsets when considering a design with only three design runs. In this instance, we assess four potential design.

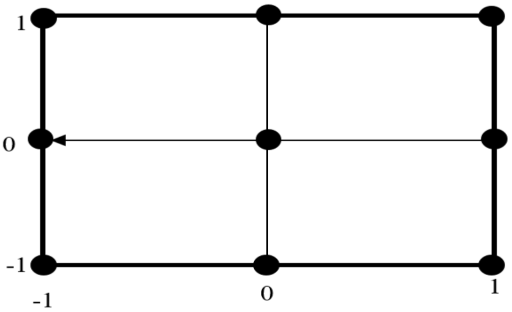


Figure 2.3: Distribution of the Candidate Set.

Matrices and assess them using the D-criterion. Figure 2.4 illustrates the experimental region with the associated candidates, and Equation 2.7 shows the four selected subsets in the matrix notation.

$$\xi_{3A} = \begin{bmatrix} -1 & -1 \\ 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad \xi_{3B} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \xi_{3C} = \begin{bmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \xi_{3D} = \begin{bmatrix} -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2.7}$$

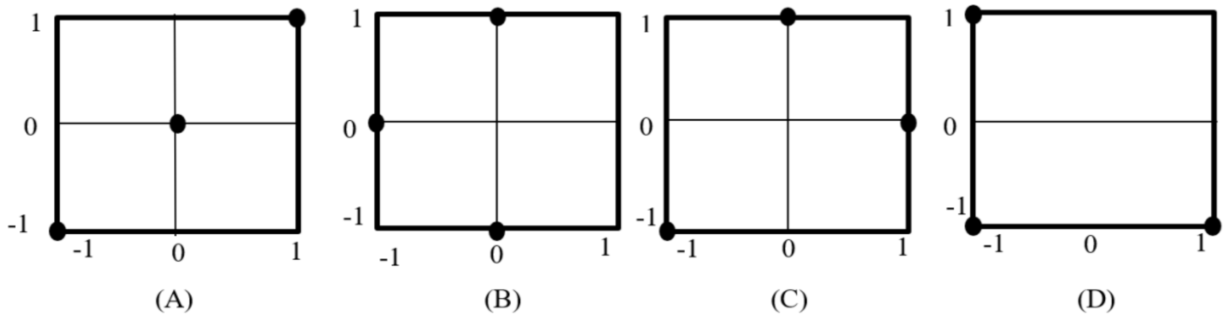


Figure 2.4: Distribution of the Design Matrix.

In order to determine which of these designs is optimum, we must first choose a model. In order to keep the example as simple as possible, we pick the following linear model:

$$y = B_0 + B_1 x_1 + B_2 x_2 + \epsilon \tag{2.8}$$

Comparing the designs depending on the D-criterion is only possible if we calculate the determinants of the information matrix($X^t X$) for each of the four designs. In this example, we only show the calculation for the subset ξ_{3B}, but the principle will be the same for all. First, we have to create the model depending design matrix X and its transpose X^t. With the linear model, we have a simple design matrix that contains the constant B_0 in the first column and the two factors in the following columns. The calculation of the information matrix is simple the multiplication of these both matrices.

$$({}^t X_B X_B) = \begin{bmatrix} 1 & 1 & 1 \\ 0 & -1 & 0 \\ -1 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & -1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \tag{2.9}$$

The determinant of this information matrix needs to be calculated next. It is recommended to apply the Sarrus rule for for $n \leq 3$. As seen in Figure 2.5, the matrix needs to be expanded in order to multiply the values and determine the subsequent diagonals from higher left to bottom right. The outcomes are totaled, and the diagonals from higher right to lower left are calculated in the same way. The matrix's determinant is the difference between the two sums. This arithmetic is demonstrated for the information matrix above in Equation 2.19. Table 2.3 shows the remaining factors.

$$|X_B'X_B| = ((3\cdot 1)+((-1)\cdot 0\cdot 0)+(0\cdot(-1)\cdot 0))-((0\cdot 1\cdot 0)+(0\cdot 0\cdot 3)+(2\cdot(-1)\cdot(-1))) = 4 \quad (2.10)$$

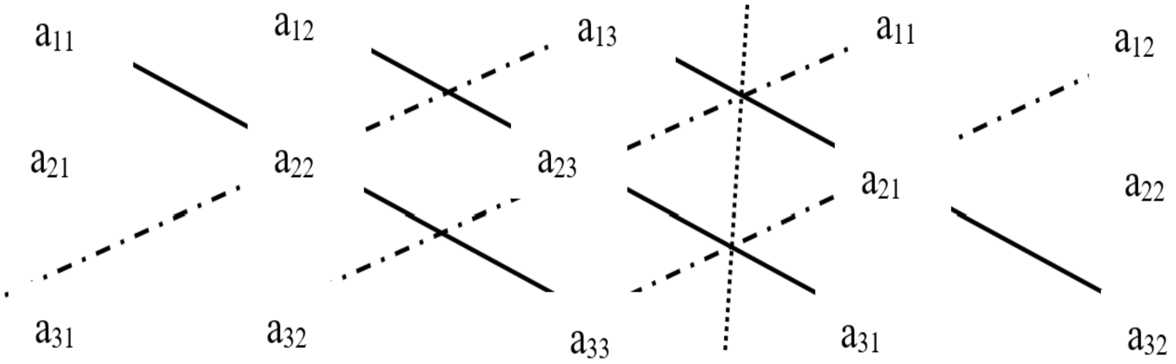


Figure 2.5: Rule of Sarrus for the Calculation of a Determinant.

It is clear from a comparison of the investigation's results that design$_3d$ is the best D-optimal design because it has the highest determinant. All of the chosen candidates for are situated at the edges of the experimental area. With a determinant of 16, all designs that examine three of the four potential corners are equally good as the one that was chosen. These designs also cover the most area over the experimental region, as Figure 2.4 illustrates.

Table 2.3: Determinants of Different Designs.

Design	Determinant
E3A	0
E3B	4
E3C	9
E3D	16

Chapter 3

D-Optimal Design as a Computer Algorithm

In this chapter, we present various exchange algorithms designed to extract an optimal subset from a given set of experiments, based on the D-optimality criterion. These algorithms optimize the selection of experiments to maximize the precision and efficiency of the parameter estimates in the model.

3.1 Exchange Algorithms

Chapter 2 describes the basic principles and criteria for constructing D-optimal designs, but does not explain the selection process itself. Due to the complexity of D-optimal designs and the huge number of possible trial combinations, computer algorithms are used in the selection process. This thesis deals only with so-called exchange algorithms. Figure 3.1 summarizes the basic steps that must be taken before such a permutation algorithm can be used.

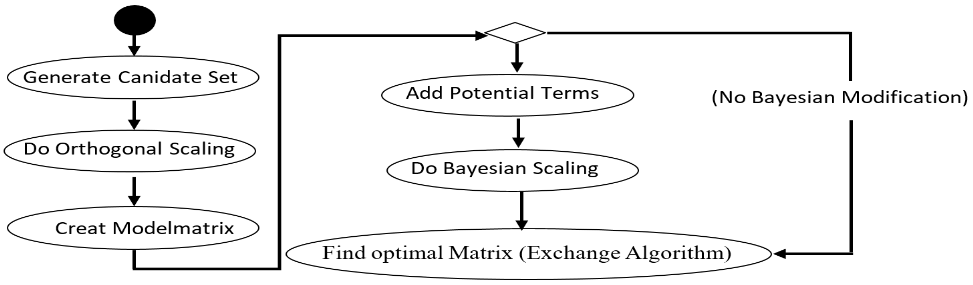


Figure 3.1: Flow chart ofthe D-Optimal Process

The permutation algorithm selects the optimal design matrix X by changing one or more points from the generated baseline and repeats that permutation until the best matrix appears to be found. Algorithms can be divided into two groups, where the 1st-order algorithm adds and removes points sequentially, and the 2nd-order algorithm performs the exchange through a simultaneous addition and removal process [Meyer and Nachtsheim (1995)]. We now explain and evaluate six different algorithms that are generally applicable but differ in the computational time used and the quality or efficiency of the plans produced.

3.1.1 General Permutation

Procedure to start the selection, we need to create an initial design with n trials in the design matrix X_n. The goal of the permutation algorithm is to remove or add points to this design matrix and determine its effect. to change Referring to Section 2.2, we call the data matrix $X_n^t X_n$ and use the following equation for the prediction variance of one candidate

$$X : d(\chi_x) = \chi_x' * (X_n' X_n)^{-1} * \chi_x \tag{3.1}$$

Adding a new experiment x to the design matrix X creates a new matrix X_{n+1} and as by [de Aguiar et al. (1995)] showed that the relationship between these two matrices can be expressed using information matrices. Equation 3.2 shows this relationship with the added use of one candidate x, and its transpose.

$$(X_{n+1}' X_{n+1}) = (X_n' X_n) + (\chi_j' * \chi_j) \tag{3.2}$$

Furthermore, the effect of the exchange may be to update the determinant of the new matrix without calculating it in the usual way. In this case, the determinant increases in proportion to the variance of the added point prediction

$$|X_{n+1}' X_{n+1}| = |X_n' X_n| * (1 + d(\chi_j)) \tag{3.3}$$

Experiment X: Elimination of Matrix The design is based on the same basic factors and is given by the following equations:

$$|X_{n-1}' X_{n-1}| = |X_n' X_n| - (\chi_i * \chi_i') \tag{3.4}$$

$$|X_{n-1}' X_{n-1}| = |X_n' X_n| * (1 - d(\chi_i)) \tag{3.5}$$

Added and removed from candidate set the selection of points is different for each algorithm and is discussed in the following sections [de Aguiar et al. (1995)] .

3.1.2 DETMAX algorithm

The DETMAX algorithm was published by Mitchell (1974) and is a typical 1st-order algorithm. Based on a random initial design of n runs, the algorithm tries to improve the determinant of the information matrix by adding or removing a point. The included experiment χ_j is the one with the largest variance of the prediction $d(\chi_j)$. As Equation 3.3 shows, this experiment leads to a maximum of the determinant. The deleted point χ_j is the point with the smallest predictive variance because changing this point decreases the determinant. If a point is added or removed first, it will be randomly selected. Such an exchange process results in a determinant greater than or equal to the previous one.

That algorithm would be the Wynn-Mitchell algorithm of 1972, but [Mitchell (1974)] modified this approach to provide more flexibility and allow project excursions. In this case, the excursion means that an $n+1$-point model cannot immediately be reduced to an n-point model, but can become an $n+2$-point model. Therefore, it may be possible to replace more than one point of the original design in a single iteration. Mitchell (1974) defined$k = 6$ like the limit of excursions. Given a model that currently has the best n points, the algorithm adds or removes a maximum of $k$points until the orbital limit is reached. The size of the generated patterns varies from$n-k$ to $n+k$. If no improvement in the determinant is found during this study, all generated models are stored in list F, which contains failure plans. This set F is used for the next expedition, which considers two different rules defined by [Mitchell (1974)] :

If D is the current model at any point of the excursion, the continuation rules are:

(i) If the number of points in D is greater than n, subtract a point if D is not in F, and otherwise add a point.

(ii) If the number of points in D is less than n, add a point. if D is not in F and subtract a point otherwise.

The following listing 3.1 shows the use of these search rules and depicts the flow of the algorithm in simple and abstract programming notation.

Algorithm 1 DETMAX algorithm by Mitchell (1974)

```
Create random design with desired number of points
Input
while excursion limit is not reached do
    if number of candidates equals number of desired runs then
        Randomize between adding or deleting a point
    else if number of candidates is bigger than number of desired runs then
        if new design is not inside set of failure designs then
            Delete candidate with lowest variance of prediction
        else
            Add candidate with highest variance of prediction
        end if
    else if number of candidates is smaller than number of desired runs then
        if new design is not inside set of failure designs then
            Add candidate with highest variance of prediction
        else
            Delete candidate with lowest variance of prediction
        end if
    end if
    if no improvement of the determinant is found then
        Save design to list of failure designs
    else
        Clear list of failure designs
    end if
end while
Output
```

For an empty set of failed plans F, the algorithm simply adds and subtracts the points as described in the first paragraph of this section. Based on the failure plans of previous iterations, the algorithm always progresses towards the n-point design, unless the F design has already failed. Then it reverses direction ([Mitchell (1974)] . If the study improves the design, the failure plans of F are removed and a new beginning is made [de Aguiar et al. (1995)] , [Mitchell (1974)]

3.1.3 Fedorov's algorithm

[Fedorov (1972)] algorithm is a simultaneous change method that always preserves the desired model size n without bias. After generating a random starting pattern, the algorithm selects point χ_i from among the design candidates that should be eliminated by point χ_j. Adding and removing a point is done in one step and can be called a real exchange. The effect of such information exchange can be demonstrated using a data matrix. In relation to equations 3.2 and 3.4, the exchange point is the simultaneous addition and subtraction.

$$(X'_{new} - X_{new}) = (X'_{old} X_{\text{old}}) - (\chi'_i * \chi_i) + (\chi'_j * \chi_j) \tag{3.6}$$

Unlike the general exchange in Section 3.1.1, [Fedorov (1972)] considered the interaction of two candidate variance functions to calculate a new determinant. Instead of adding and subtracting the variance of the two-point forecast according to formulas 3.3 and 3.5, he defined the so-called "delta" function that changes the determinant of a matrix as follows:

$$|X'_{new} X_{new}| = |X'_{old} X_{old}| * (1 + \Delta(\chi_i, \chi_j)) \tag{3.7}$$

Calculating the value for the pair χ_i and χ_j uses A and the variance of the scores and the prediction of a combined variance function called $d(\chi_i, \chi_j)$.

$$\Delta(\chi_i, \chi_j) = d(\chi) - [d(\chi_i) d(\chi_j) - (d(\chi_i, \chi_j))^2] - d(\chi_i) \tag{3.8}$$

$$d(\chi_i, \chi_j) = \chi'_i * (\chi'_n \chi_n)^{-1} * \chi_j = \chi'_j * (\chi'_n \chi_n)^{-1} * \chi_i \tag{3.9}$$

The main idea of Fyodorov's algorithm is to calculate the value of Δ for all possible pairs (χ_i, χ_j) and choose the one with the maximum value . The point is χ_i taken from the currently selected model, and χ_j can be taken either from the remaining points or from the entire set of candidates. Considering only the remaining points is called an exhaustive search, which avoids overlapping points in the design. With an incomplete search, it is possible to select an exam that must be taken twice. As shown in Equation 3.7, the points with the largest value of Δ increase the determinant the most. If more than one pair with the same Δ- value is found, the algorithm randomly chooses from among them. If pairs with a positive Δ- value are found, the algorithm exchanges points and updates the information and variance matrix. Sometimes the algorithm finds pairs that increase the determinant so little that no significant difference is achieved. To avoid these pair-change algorithms, [Fedorov (1972)] defined an input threshold and broke the algorithm when the maximum value of Δ is less than the input, where 10^-6 is the common value of the input [de Aguiar et al. (1995)] , [Fedorov (1972)] . The general description of the algorithm is given in the "Algorithm 2".

Algorithm 2 Fedorov Algorithm (1972)

```
Create random design with desired number of points
Input
while couples with positive delta are found do
    for design point $\chi_i$ to design point $\chi_n$ do
        Calculate variance of prediction $d(\chi_i)$ for this design point
        for support point $\chi_j$ to support point $\chi_N$ do
            Calculate variance of prediction $d(\chi_j)$ for this support point
        end for
        Calculate variance function $d(\chi_i, \chi_j)$ for this couple
        Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
        Check if maximum delta and save couple
    end for
    if maximum delta is positive then
        if more than one couple with same maximum delta then
            Select couple randomly
        end if
        Exchange selected point $\chi_i$ with $\chi_j$
        Update information and dispersion matrix
        Reset maximum delta
    end if
end while
Output
```

3.1.4 Modified Fedorov Algorithm

[Cook and Nachtsheim (1980)] compared different algorithms for creating an accurate D-optimal model and devised their own algorithm based on the basic Fedorov algorithm of 1972. The normal Fedorov algorithm calculates the Δ values of all possible exchange pairs during one iteration, but uses only one of the values to make the exchange. This calculation is expensive to use. In a modified version by [Cook and Nachtsheim (1980)], each iteration of the standard algorithm is divided into [n] steps, one for each design support at the beginning of the iteration." Ordered design matrix, the algorithm. starts at the first supportχ_i and calculates the values of Δ for all possible pairs in it at a fixed reference point. Once the best replacement for that point is found, the design is updated and the next support point is proposed to replace. In other words, one iteration of the standard Fedorov algorithm is modified to vary up to n design points if the determinant should increase ([Cook and Nachtsheim (1980)] , [Atkinson and Donev (1989)] . This behavior is illustrated in the "Algorithm 3".

The difference between the two approaches can be explained with a simple example. Given a desired model with $n = 5$ trials and a candidate set with $N = 20$ trials, $n * N = 100$ possible pairs can be offered for exchange. Each iteration of Fedorov's algorithm calculates these 100 Δ - values for all possible pairs and uses only one of them to switch. In contrast, the modified version of the algorithm starts from the first drawing point and calculates only 20 Δ-values for possible pairs including that point. After that, the exchange takes place and the variance matrix must be updated. The algorithm continues to the next design point and recalculates the 20 Δ-values. In general, both algorithms compute 100 values in one iteration, but the modified version exchanges up to 5 points during this operation.

A study by [Atkinson and Donev (1989)] and the data presented in the following Section 3.3 show that the modified approach can be twice as fast as the standard Fedorov.

Algorithm 3 Modified Fedorov Algorithm by Cook & Nachtsheim (1980)

```
Create random design with desired number of points
Input
while couples with positive delta are found do
    for design point $\chi_i$ to design point $\chi_n$ do
        Calculate variance of prediction $d(\chi_i)$ for this design point
        for support point $\chi_i$ to support point $\chi_N$ do
            Calculate variance of prediction $d(\chi_j)$ for this support point
        end for
        Calculate variance function $d(\chi_i, \chi_j)$ for this couple
        Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
        Select couple with maximum delta
    end for
    if maximum delta is positive then
        if more than one couple with same maximum delta then
            Select couple randomly
        end if
        Exchange selected point $\chi_i$ with $\chi_j$
        Update information and dispersion matrix
        Reset maximum delta
    end if
end while
Output
```

Algorithm and creates design with a comparable efficiency. The additional time needed to update the dispersion matrix after each exchange is adjusted by the profit of the multiple exchanges.

3.1.5 k-Exchange algorithm

Comparing the standard Fedorov (1972) algorithm and Mitchell's DETMAX (1974), Johnson and Nachtsheim (1983, p. 274) found that the points selected for deletion by Fedorov's algorithm are normal not those with the lowest predicted variance, but "the frequency of deleted points can be characterized as biased towards the lower variance classes". Simply put, instead of considering all candidates or only the candidates with the smallest predictive variance, the set of k points with the smallest variance should be selected. Similar to the modified Fedorov algorithm, the iteration is now divided into k steps. At each of these k steps, the values of Δ are computed and the corresponding pair is exchanged if the determinant should be increased [Johnson and Nachtsheim (1983)]).

Similarity to the algorithms described above can be detected when we define k. value For k = 1, the algorithm is similar to the Wynn-Mitchell algorithm in Section 3.1.2, and for k = n it becomes a modified Fedorov algorithm [Cook and Nachtsheim (1980)] . Choosing the value of K is difficult and largely depends on the problem. A common value, also suggested by [Johnson and Nachtsheim (1983)], is k=3 or k=4. [Meyer and Nachtsheim (1995)] later proposed to choose a value with the following condition:

$$k \leq \frac{n}{4} \tag{3.10}$$

Algorithm 4 Modified Fedorov Algorithm by Cook & Nachtsheim (1983)

```
Create random design with desired number of points
Input
while couples with positive delta are found do
    Calculate variance of prediction $d(\chi_i)$ for all design points
    select $k$ design points with lowest variance of predection
    for design point $\chi_i$ to design point $\chi_k$ do
        Calculate variance of prediction $d(\chi_i)$ for this design point
        for support point $\chi_i$ to support point $\chi_N$ do
            Calculate variance of prediction $d(\chi_j)$ for this support point
            Calculate variance function $d(\chi_i, \chi_j)$ for this couple
            Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
            Select couple with maximum delta
        end for
        if maximum delta is positive then
            if more than one couple with same maximum delta then
                Select couple randomly
            end if
            Exchange selected point $\chi_i$ with $\chi_j$
            Update information and dispersion matrix
            Reset maximum delta
        end if
    end for
end while
Output
```

Continuing the example with $n = 5$ and $N = 20$, k-swap speeds up the selection process by reducing the total number of calculated Δ-values. Similar to the modified Fedorov procedure, each iteration of the basic Fedorov algorithm is divided into Δ- calculations for each design point. However, depending on the specified $k-value$, the algorithm only considers a few design points. When $k = 3$, the k-swap algorithm calculates 20 Δ-values three times and makes up to three swaps. This approach has higher computational efficiency compared to the modified Fedorov procedure. Especially for a large dataset with a large n value, defining the value of k has an impact. The quality of the design is not always as good as the standard Fedorov algorithm, but considering the required computation time, the algorithm gives good results. A detailed analysis of the effectiveness and quality of the plans is presented in section 3.3.

3.1.6 kl-Exchange Algorithm

The kl-exchange algorithm, a variant of the original Fedorov algorithm introduced in 1972, is a typical rank-2 algorithm developed by Atkinson and Donev in 1989. This method streamlines the process by narrowing down the selection of support and exchange points before computing the Δ-values for all potential pairs. It mirrors the k-exchange procedure in its utilization of k points with the least variance in prediction. Furthermore, it exclusively considers candidates with the highest prediction variance among the support points. Setting$k = n$ and $l = N$aligns the algorithm with the standard Fedorov procedure, while initiating it with $k < n$ and $I < N$ reduces the number of computed Δ-values, thus accelerating the algorithm. Similar to the conventional Fedorov approach, the kl-exchange identifies points with the highest Δ-value and executes a single exchange. Despite its name, the kl-exchange is distinct from the k-exchange by [Cook and Nachtsheim (1980)], relying instead on the fundamental Fedorov algorithm and refraining from multiple exchanges within a single iteration. The sole similarity lies in the shared concept of selecting points with the lowest prediction variance. [Atkinson and Donev (1989)] proposed two alterations to this algorithm. The first modification resembles the essence of the modified Fedorov algorithm by [Cook and Nachtsheim (1980)], enabling multiple exchanges per iteration. When $l = 1$ is selected, this modified kl-exchange transforms into either the modified Fedorov algorithm with kn or the k-exchange with $k < n$. The second modification alters the selection criteria for the k and I points, opting for random selection from the design and support points instead of variance-dependent choices [Atkinson and Donev (1989)].The "Algorithm 5" showcases the basic version of the algorithm without any adjustments.

Algorithm 5 kl-Exchange Algorithm by [Atkinson and Donev (1989)].

```
Create random design with desired number of points
Input
while couples with positive delta are found do
    Calculate variance of prediction $d(\chi_i)$ for all design points
    Select $k$ design points with lowest variance of prediction
    Calculate variance of prediction $d(\chi_j)$ for all support points
    Select support points with highest variance of prediction
    for design point $\chi_1$ to design point $\chi_k$ do
        Calculate variance of prediction $d(\chi_i)$ for this design point
        for support point $\chi_1$ to support point $\chi_l$ do
            Calculate variance of prediction $d(\chi_j)$ for this support point
            Calculate variance function $d(\chi_i, \chi_j)$ for this couple
            Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
            Check if maximum delta and save couple
        end for
    end for
    if maximum delta is positive then
        if more than one couple with same maximum delta then
            Select couple randomly
        end if
        Exchange selected point $\chi_i$ with $\chi_j$
        Update information and dispersion matrix
        Reset maximum delta
    end if
end while
Output
```

Compared with the ordinary Fedorov calculation, the kl-exchange decreases the sum of calculated Δ-values between each cycle. Our case with $n = 5, N = 20$ and $k = 1 = 3$ leads to the calculation of as it were $k * 1 = 3 * 3 = 9$ Δ-values. Compared with the 100 couples of the ordinary Fedorov calculation, this kl-approach speeds up the calculation in an viable way. In fact, we got to watch that the sum of performed emphasess of the kl-exchange can be higher because not all couples are considered for an trade, and so the calculation needs to perform more circles. But in common, the kl-exchange calculation makes plans exceptionally quick and in most cases gives satisfactory comes about considering the D-optimality.

3.1.7 Modified kl-Exchange Algorithm

Based on the past encounters of the company metrics AB and the information picked up amid the usage prepare of diverse calculations, a adjusted adaptation of the kl-exchange calculation is created within the display proposal. The primary straightforward alter within the calculation is to amplify the k-value in the event that the k chosen focuses have the same change of forecast and a comparative $(k + 1)$ the steem exists. Employing a arbitrary selection if more than k

candidates have the same change can moreover be a arrangement but isn't as foolproof as this one. The moment adjustment tries to avoid the kl-exchange calculation from trading a few couples which are not the leading choice by decreasing the list of bolster focuses. Beginning with $l = N$ focuses within the pool of conceivable back focuses, the calculation expels focuses depending on diverse criteria. By and large, an trade ought to as it were be performed on the off chance that the fluctuation of forecast $d(\chi_j)$ of the bolster point χ_j, is higher than the change of expectation $d(\chi_i)$ of the evacuated plan point χ_i. A moment measure to decrease the list of bolster point is created subordinate on the normal change of forecast from the final cycle. As it were the bolster focuses with the next change of expectation than the normal change are considered. In this case, the normal fluctuation of expectation is calculated with the utilize of all conceivable candidates. On the off chance that the calculation does not discover a most extreme Δ-value among these conceivable best support points, the remaining support points are checked. The total calculation is appeared in the "Algorithm 6".in This calculation is troublesome to compare to the past approaches since the sum of design and back point considered for an trade is problem-dependent. We don't have settled values for I and k and the criteria for the determination of the focuses can change amid the method. But in common, the altered kl-exchange is comparable to the typical kl-exchange, considering the computational efficiency. The calculations require a few more time since we now and then alter the fluctuation criteria and calculate the Δ-values once more to discover the leading trade. But in general, the calculation is comparable to the typical kl-exchange. But considering the quality of the produced plans, this calculation leads to superior results and in most cases too beats the k-exchange method. Of course, the comes about don't have the same quality as the plans made with the Fedorov calculation. But with an allowable computing time, we make great D-optimal designs. In common, the comparison of the quality of the plans is troublesome since most of the calculations have distinctive quality and shortcomings but ought to be appropriate for all issue details. This issue is examined in detail in section 3.3.

Algorithm 6 Modified kl-Exchange Algorithm

```
Create random design with desired number of points
Input
while couples with positive delta are found do
    Calculate variance of prediction $d(\chi_i)$ for all design points
    if more than $k$ design points have same lowest variance of prediction then
        Extend given $k$
    end if
    Select $k$ design points with lowest variance of prediction
    Calculate variance of prediction $d(\chi_j)$ for all support points
    Calculate average variance of prediction over whole candidate set
    Select support points with higher variance of prediction than the average value
    for design point $\chi_1$ to design point $\chi_k$ do
        Calculate variance of prediction $d(\chi_i)$ for this design point
        for support point $\chi_1$ to support point $\chi_l$ do
            Calculate variance of prediction $d(\chi_j)$ for this support point
            if variance of the support point is higher than the variance of the design point then
                Calculate variance function $d(\chi_i, \chi_j)$ for this couple
                Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
                Check if maximum delta and save couple
                if maximum delta is negative then
                    Select $N - l$ support points that have not been checked
                    for design point $\chi_1$ to design point $\chi_k$ do
                        Calculate variance of prediction $d(\chi_i)$ for this design point
                        for support point $\chi_1$ to support point $\chi_{N-1}$ do
                            Calculate variance of prediction $d(\chi_j)$ for this support point
                            Calculate variance function $d(\chi_1, \chi_j)$ for this couple
                            Calculate delta function $\Delta(\chi_i, \chi_j)$ for this couple
                            Check if maximum delta and save couple
                        end for
                    end for
                    if maximum delta is positive then
                        if more than one couple with same maximum delta then
                            Select couple randomly
                        end if
                        Exchange selected point $\chi_i$ with $\chi_j$
                        Update information and dispersion matrix
                        Reset maximum delta
                    end if
                end if
            end if
        end for
    end for
end while
Output
```

3.2 Generation of the Start Design

Another strategy to build D-optimal plans was created by [Dykstra (1971)]some time recently the common trade calculations from segment 3.1 were utilized. Rather than starting with a plan with the required measure n, [Dykstra (1971)] begun with an purge plan and included consecutively the plan focuses with the most elevated change of forecast to maximize the determinant. This arrangement gives destitute comes about considering the productivity of the plan. But the approach can be utilized as a quick strategy for producing a beginning plan for other trade calculations. Compared with a randomized beginning plan, a successively chosen one diminishes the number of emphasess of an calculation since the focuses within the plan are as of now a preselection.

[Galil and Kiefer (1980)] used the idea of a sequential start design for [Mitchell (1974)] DETMAX algorithm and changed the start design to contain some random points in the beginning. Thereafter, the design is sequentially filled with the best support points. The number of the t random points was defined by [Galil and Kiefer (1980)] to be 'considerable smaller than k for moderate k [...]'. [Atkinson and Donev (1989)] referred to this topic by using a sequential start design for the ki-exchange algorithm. They defined the number of random selected points t as a randomly chosen integer with the following condition:

$$0 \leq t \leq min([\frac{1}{2}P], N-1) \tag{3.11}$$

.

where $\frac{1}{2}P$ denotes the integer part of the fraction and p is the number of used factors.

3.3 Comparison of Exchange Algorithms

Generally, a comparison of the calculations should be done depending on two fundamental criteria. The primary one is the utilized computing time to form a D-optimal plan, and the moment one is the quality of the plan. This quality or productivity of the plan can be assessed by the criteria taken from section 2.3. In this investigation we utilize the D-Optimality or its analog, the determinant to assess the ideal plans. Comparing other criteria, just like the condition number or the G-Efficiency, can lead to other comes about than the ones appeared in this segment. The base for the given information in this comparison is an examination of the four calculations implemented at Umetrics AB. The case ponder was performed with an interaction demonstrate with 5, 7 or 9 components. The program MO is utilized to make the specified plans and calculates their productivity, the logarithm of the determinant. Amid the calculation of 10 plans for each demonstrate and calculation, the computing time is measured and the cruel esteem is calculated. The collected information can be found in table 3.1.

Table 3.1: Comparison of Different Algorithms.

	5 factors		**7 factors**		**9 factors**	
	time/ms	log(Det)	time/ms	log(Det)	time/ms	log(Det)
Fedrov	21.9	**20.47**	424.8	**43.74**	6887.2	75.33
mod.Fedrov	21.8	**20.47**	286.7	**43.74**	3535.4	**75.36**
k.exchange	**10.9**	**20.47**	**84.6**	43.71	**669.2**	74.83
mod.kl-exchange	20.3	**20.47**	126.6	43.72	931.7	74.99

Referring to the description of the algorithms in Section 3.1, we show that some modifications reduce the number of calculations before making the exchange. For example, Fedorov's algorithm calculates the value of Δ for all possible pairs and makes only one exchange depending on this data. Due to several factors, this approach is very expensive to use. A modified version of [Cook and Nachtsheim (1980)] reduces the number of calculations by making up to n exchanges during each iteration and speeds up the algorithm by using more than one value. Table 3.1 shows that the average computation time of the modified Fedorov algorithm can be up to 50% faster than the standard 1972 algorithm. Especially for large input data, the difference between the two algorithms becomes apparent. Other algorithms, such as the k-exchange of [Johnson and Nachtsheim (1983)] or a modified kl-exchange, are much faster for generating D-optimal models and can be used for a wide variety of factors. For example, k-swap can process the interaction model with nine factors in Table 3.1 ten times faster than the basic Fedorov procedure. Mitchell's algorithm was not applied in this thesis but was studied and compared by [Johnson and Nachtsheim (1983)]. The evaluated data from different models and factor settings showed that the computation time of the DETMAX algorithm is comparable to the standard k-swap.However, the fast algorithm is not suitable if the generated model lacks the desired quality or D-optimality . Therefore, we have to compare algorithms according to time consumption and the quality of the design created. At this stage, it should be understood that the measurement data in Table 3.1 are difficult to evaluate. All algorithms can stay at the so-called local optimum and stop the switching process even if the best model has not yet been found. As the factors or model terms increase, so does the number of local optima. Therefore, most D-optimal models are different, even if they rely on the same algorithm. In addition, features randomly selected during the selection process affect the final result. Table 3.1 shows only the best design performance of the ten generated options. In general, it is recommended to create more than one template and choose the best possible one among the templates. The determinant of the information matrix (X'X) for a small test, as in Table 3.1 with five factors, has the same value for all algorithms. In this case, it doesn't matter which algorithm is used, they all create at least one model out of ten with the same efficiency. As the number of factors increases or the model becomes more complex, Fedorov and modified Fedorov usually produce the best models, but the choice of the best algorithm is usually problem-specific. For example, the modified kl exchange produces better models than Fedorov or modified Fedorov in some cases, or is at least comparable. This algorithm is indeed much faster than Fedorov's version and should be preferred.

The conclusions in the previous paragraph are based on the data presented in Table 3.1 and other random samples. They are all based on versions of the algorithms implemented in the MODDE software. Therefore, this data is only an indicator that gives assumptions about which algorithm to use according to different goals. Furthermore, only four implemented algorithms are considered. The literature does not compare all algorithms under the same test conditions. Therefore, the selection of four algorithms is based on two different comparisons. First of all. by [Cook and Nachtsheim (1980)] and analyzed Mitchell's DETMAX (1974), the Fedorov procedure, the modified Fedorov procedure, the Wynn-Mitchell algorithm, and some other trivial algorithms. The article states that the best models are generated using Fedorov or modified Fedorov, but if fast generation is more important, the Wynn-Mitchell algorithm should be used.

Another evaluation was done by [Johnson and Nachtsheim (1983)] and compared the Fedorov algorithm, the modified Fedorov algorithm, the k-swap procedure, DETMAX and the Wynn-Mitchell algorithm. The result of this work is the advice of the authors to use the Wynn-Mitchell algorithm or its rough equivalent procedure, k-swap. Based on these findings and individual publications of the algorithms, the four algorithms presented in Table 3.1 are implemented in MODDE. Our small case study agrees with the advice of [Johnson and Nachtsheim (1983)] and [Cook and Nachtsheim (1980)], but a separate complexity analysis comparing all algorithms against the same criteria is required for a reliable comparison. from various problem formulas. However, such studies are outside the scope of this thesis.

In general, finding the optimal combination of computing time and the efficiency of the generated design is a complex topic and cannot be concluded with strict advice on which algorithm to use. In short, the modified Fedorov algorithm is similar to the standard Fedorov procedure in terms of design quality, but requires less computation time. In fact, both algorithms require a lot of computation when the input data becomes large. Therefore, the modified Fedorov algorithm of [Cook and Nachtsheim (1980)] should be used for small models. In larger models, a simple k-transform or a modified kl-transform is recommended. The generated plans are slightly worse in some cases, but in terms of calculation time, this is the best choice. If the experimenter wants to generate accurate D-optimal designs and has the ability to evaluate the problem formulation with different algorithms and parameters, this is the recommended way to find the optimal design.

In general, we do not recommend building an exact D-optimal model, because choosing the right model is difficult and must be handled carefully. To return to Section 2.8, we repeat the same case study of Table 3.1 using the Bayesian variant and try to confirm the above results. The addition of possible conditions makes the plans more model-dependent, and thus the efficiency of the plans is slightly below that of the standard D-optimization process in Table 3.1. A comparison of both studies shows that the outstanding values follow the same regularity and prove the proposals to choose the right algorithm.

Table 3.2: Comparison of Different Algorithms (Bayesian Modification).

	5 factors		**7 factors**		**9 factors**	
	time/ms	log(Det)	time/ms	log(Det)	time/ms	log(Det)
Fedrov	51.6	**19.46**	798.8	**43.41**	11343.4	74.6
mod.Fedrov	35.8	**19.46**	295.6	**43.41**	4303.0	**74.77**
k.exchange	**21.9**	**19.46**	**129.5**	43.16	**915.5**	74.32
mod.kl-exchange	31.3	**19.46**	215.7	42.71	1394.4	74.43

Chapter 4

Testing and Results

The construction of experimental designs is facilitated by the use of specific software. The interpretation of experimental designs requires numerous calculations and numerous graphs. In this chapter, we present our software designed to simplify and improve the design of mixture plans for any number of components. Our software offers a comprehensive suite of features to enable you:

- **Flexible Component Handling:**Easily define components, their properties, and any relevant constraints.
- **Powerful Optimization Engine**Uncover the ideal combination of components that yield your desired mixture characteristics
- **Intuitive User Interface:**Navigate effortlessly through the software's functions to streamline your workflow.
- **Advanced Visualization Tools:**Generate insightful graphs that visually represent the relationships between component ratios and mixture behavior.
- **Detailed Reports:**Generate comprehensive reports documenting your design process, results, and recommendations.

4.1 Benefits of Utilizing Our Software

By leveraging our software, you can achieve:

- **Streamlined Design Process**:Effortlessly define your mixture requirements and explore different scenarios.
- **Reduced Complexity:** Manage mixtures with any number of components efficiently.

- **Data-Driven Decisions:** Gain valuable insights through intuitive visualizations, guiding you towards optimal formulations.
- **Enhanced Efficiency:** Automate calculations and generate reports, freeing up your time for analysis and interpretation.

4.2 Learning Made Easy

Our software goes beyond just functionality, providing valuable learning resources:

- **Pre-loaded Examples:**Explore practical examples demonstrating how to utilize the software for diverse mixture design problems.
- **Interactive Tutorials:** Gain hands-on experience through interactive tutorials that guide you through the software's features

4.3 Beyond This Chapter

This chapter will provide a comprehensive foundation of mixture design principles. We'll explore key concepts, the theoretical framework, and most importantly, showcase the capabilities of our software through practical examples demonstrating its versatility for mixtures with any number of components. By the end, you'll be well-equipped to confidently design optimal mixtures for your specific needs, regardless of complexity.

4.4 Environment and Work Tools

we have used the laptop:

- **Processor:** 11th Gen Intel(R) Core(TM) i5-1145G7 @ 2.60GHz 1.50 GHz
- **RAM:** 8GB.
- **System:** 64-bit system, x64 processor.

We carried out our experiments using:

- **Python 3.6:** which is an open source object-oriented programming language [Lutz, 2013].
- We developed our models and carried out the experiments using the GPU and CPU computing resources provided by Google Colaboratory [Bisong, 2019]. More commonly

known as "Google Colab" or simply "Colab" which is a free service hosted by Google, based on the Jupyter environment, for teaching and research on machine learning.

In addition, the machine learning process was carried out using a set of Python packages such as:

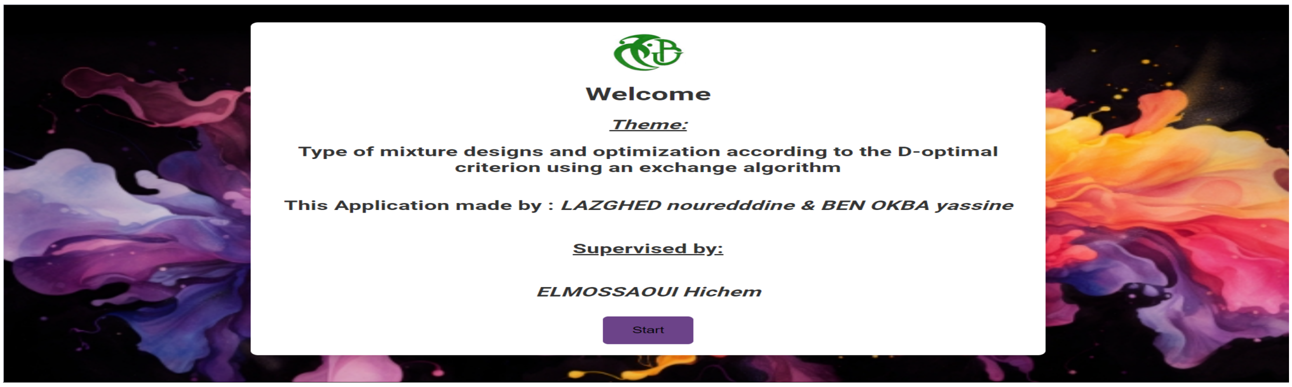
- **Scikit-learn:** which is a machine learning library that offers a wide variety of algorithms.
- **Flask:** which is a small and lightweight Python web framework that provides useful tools and features to make it easier to create Python web applications. It offers developers some flexibility and is a more accessible framework for developers as you can quickly create a web application using a single Python file.
- **Pandas:** is an open-source data analysis and manipulation library for Python. It provides data structures and functions needed to work with structured data seamlessly, particularly useful for data cleaning, transformation, and analysis.
- **NumPy:** is a fundamental library for numerical computing in Python. It provides support for arrays, matrices, and many mathematical functions to operate on these data structures efficiently.

Other libraries were invoked during our experiments, including Numpy, Pandas, Matplotlib.

4.5 Startup

- **Selection screen:** This interface element allows users to choose from a predefined set of options.
- **Type selection:**The available options correspond to different categories, such as product types, service types, or data categories.
- **Input field:**This screen element functions as an input field where users provide their selection by clicking on their desired option.

To launch our web application on port 5003, navigate to the following URL: http://127.0.0.1 (localhost). To access on the application, simply click on `http://127.0.0.1:5003`, which will direct you to the home page (Figure 4.1). This page serves as the entry point to our web application and features a 'Start' button, allowing users to commence their interaction with the application.



Welcome

Theme:

Type of mixture designs and optimization according to the D-optimal criterion using an exchange algorithm

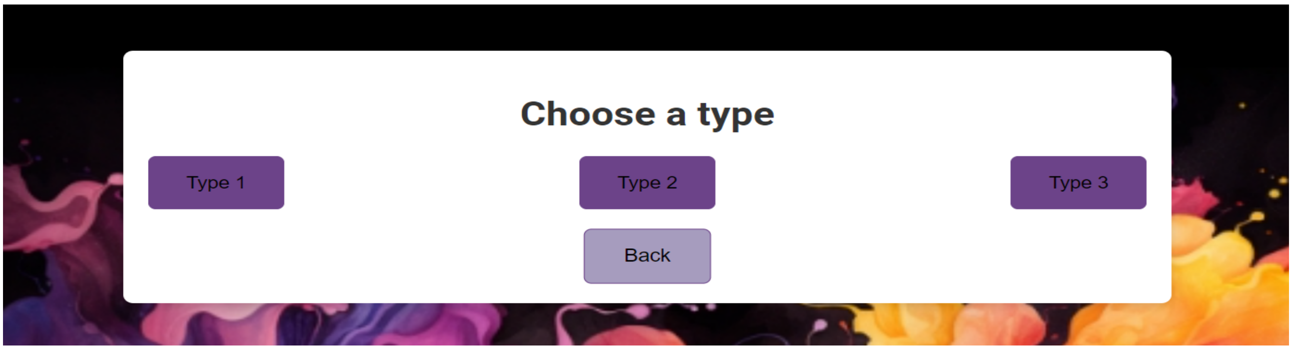
This Application made by : *LAZGHED noureddine & BEN OKBA yassine*

Supervised by:

ELMOSSAOUI Hichem

Figure 4.1: Home page of the application

This interface given by (Figure 4.2) features a selection screen where users are prompted to choose from three options: Type 1, Type 2, and Type 3. Centrally aligned buttons make it easy for users to interact and proceed with their selection. Every step features a 'Back' button that takes the users to the previous one



Choose a type

Figure 4.2: Type selection

We will describe the use of the application web by relying on the following example :

4.6 Example 1 of the Type I(Linear)

Consider the example given in chapter 1 section 5

Our focus is on examining the low-temperature behavior of a ternary mixture, with a required precision of 0.5°C for the cold hold. We have analyzed various mixture design plans to develop first, second, and third-degree models. These plans incorporate experimental points situated at the vertices, along the edges, and at the center of the study domain. In this particular example, we investigate a type 1 mixture design, which assumes no interactions between the components.

4.6.1 Hypothesis of a First-Degree Model

The first-degree model assumes that the response depends linearly on the proportions of the components. The mathematical expression of this model is:

$$Y = b_1X_1 + b_2X_2 + b_3X_3$$

1. Y is the response (cold hold)

2. b_1, b_2, and b_3 are the model coefficients

3. X_1, X_2, and X_3 are the proportions of the components

Table 4.1: The matrix of related experiences:

mixture	x_1	x_2	x_3	Cold resistance (°C)
1	1	0	0	-40.5
2	0	1	0	-12.5
3	0	0	1	-19

- **Model Equation**

$$y = -40.5x_1 - 12.5x_2 - 19.0x_3.$$

- **Model Validation** The validity of the first-degree model is tested with a new experiment.

Table 4.2: The matrix of related experiences:

mixture	x_1	x_2	x_3	Cold resistance (°C)
4	1/3	1/3	1/3	-26.9

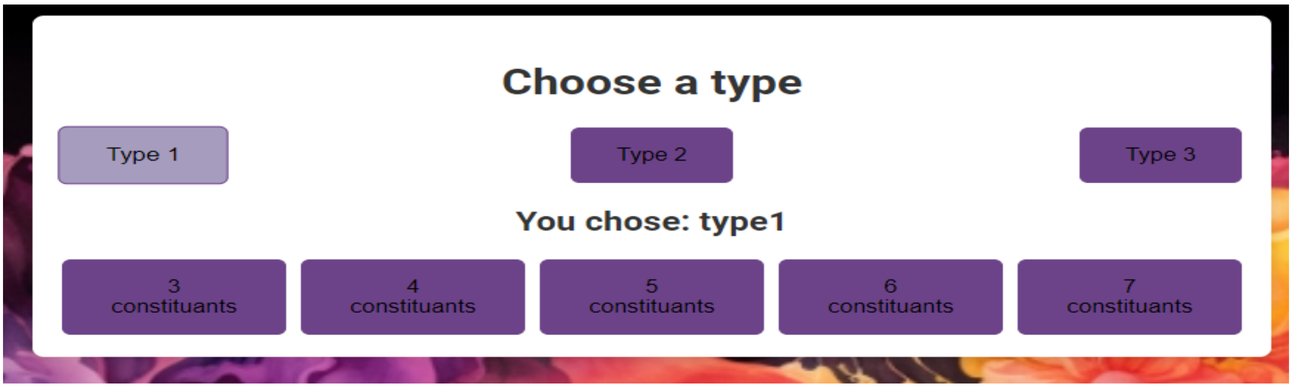
- **Observed Response** The observed response is Yobs = -26.9.

- **Calculated Response** The calculated response is ycalc = -24.

- **Results** Since the error is greater than the precision (0.5°C), the test is not verified, and the first-degree model is not validated.

4.6.2 Description of the web application interfaces For Type I Linear

The picture (Figure 4.3) allow to choose the type.

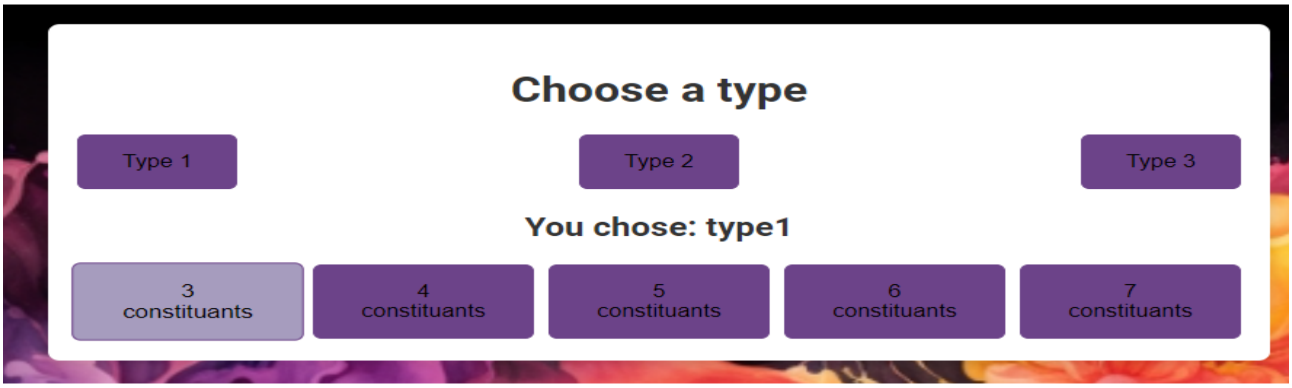


Choose a type

You chose: type1

Figure 4.3: Chosen type 1

After choosing a Type (Type 1, Type 2, Type 3) this interface (Figure 4.4) :allows users to select a the number of constituents for the selected type, ranging from 3 to 7. The user's current selection is displayed below the type options, confirming their choice.

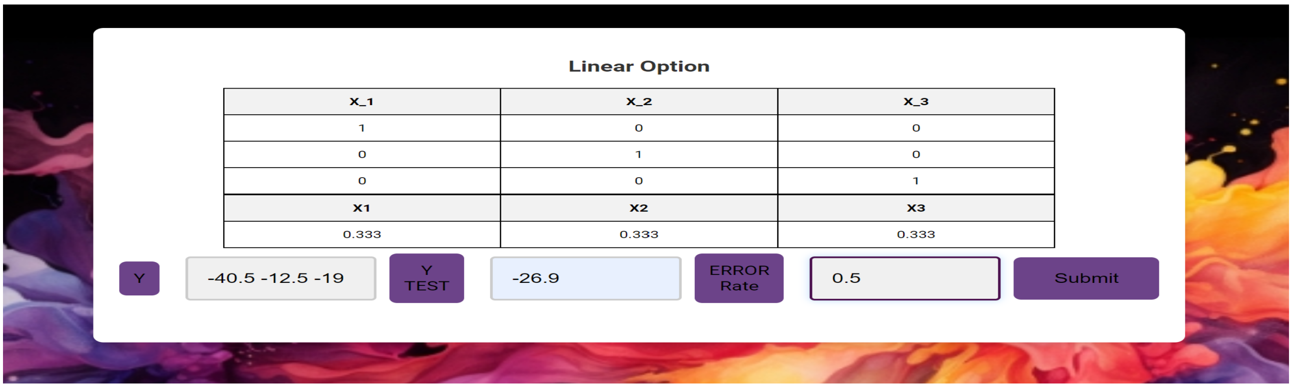


Choose a type

You chose: type1

Figure 4.4: Chosen a three constituents

After choosing a Type (Type 1, Type 2, Type 3) and then selecting the number of constituents (ranging from 3 to 7) for the chosen type. Additionally, the user can choose between two options: Linear or Quadratic option.



Linear Option

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1
X1	**X2**	**X3**
0.333	0.333	0.333

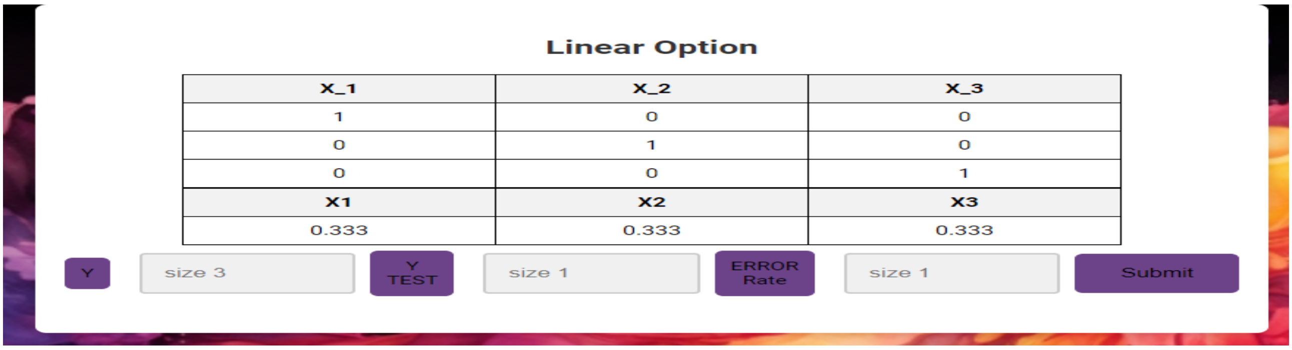
Y: -40.5 -12.5 -19

Y TEST: -26.9

ERROR Rate: 0.5

Figure 4.5: Chosen a linear model

This interface (Figure 4.6) displays the "Linear Option," showing a table with columns X_1, X_2, and X_3, each containing binary values and Testing vector values of X1, X2, and X3. Below the table, there are buttons and input fields labeled Y, Y TEST, ERROR Rate, and Submit button, as well as size indicators for the Y and Y TEST inputs.



Linear Option

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1
X1	X2	X3
0.333	0.333	0.333

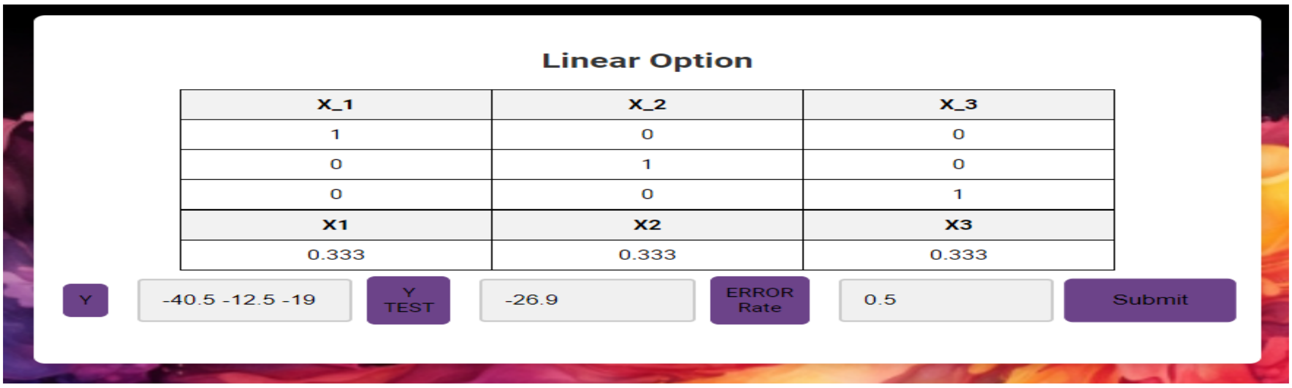
size 3

size 1

size 1

Figure 4.6: Enter the values for type 1

This picture shows the values of Y and Y TEST and the Eroor Rate of our example



Linear Option

X_1	X_2	X_3
1	0	0
0	1	0
0	0	1
X1	X2	X3
0.333	0.333	0.333

-40.5 -12.5 -19

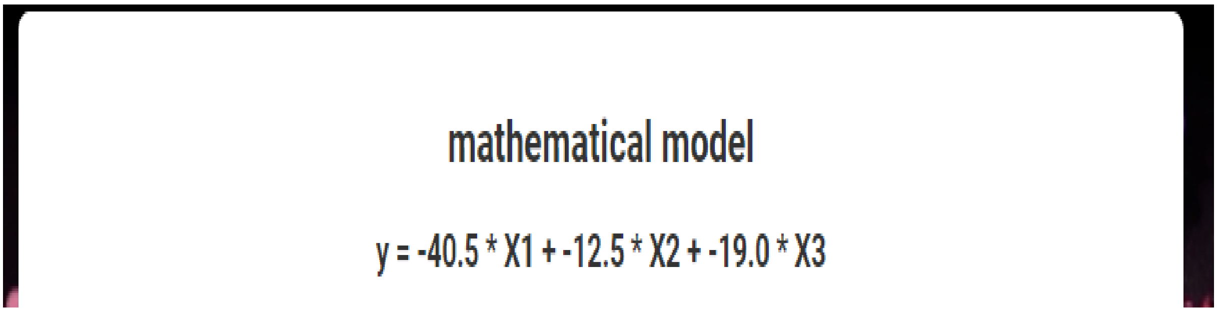
-26.9

0.5

Figure 4.7: Values have been entred for type 1

4.6.3 Display Linear Results

First, the application display the mathematical model

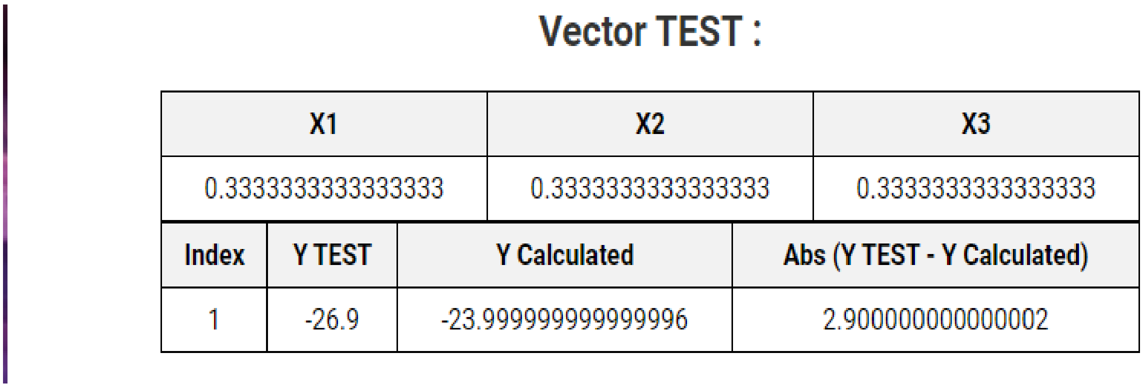


mathematical model

y = -40.5 * X1 + -12.5 * X2 + -19.0 * X3

Figure 4.8: Mathematical model

Second ,the application display the difference between y test and y calculated and compare it with the error rate



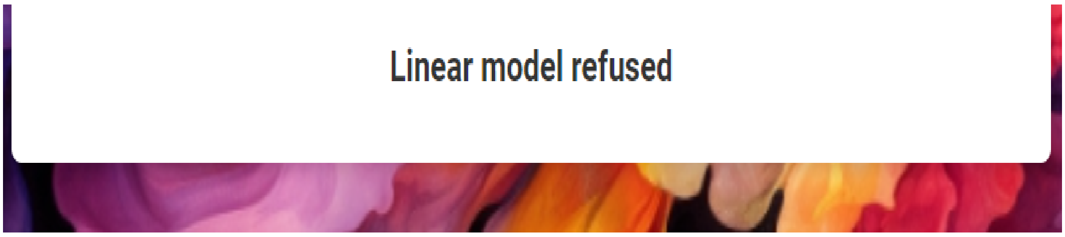
Vector TEST :

X1	X2	X3
0.3333333333333333	0.3333333333333333	0.3333333333333333

Index	Y TEST	Y Calculated	Abs (Y TEST - Y Calculated)
1	-26.9	-23.999999999999996	2.900000000000002

Figure 4.9: Table displaying results of Y calculated

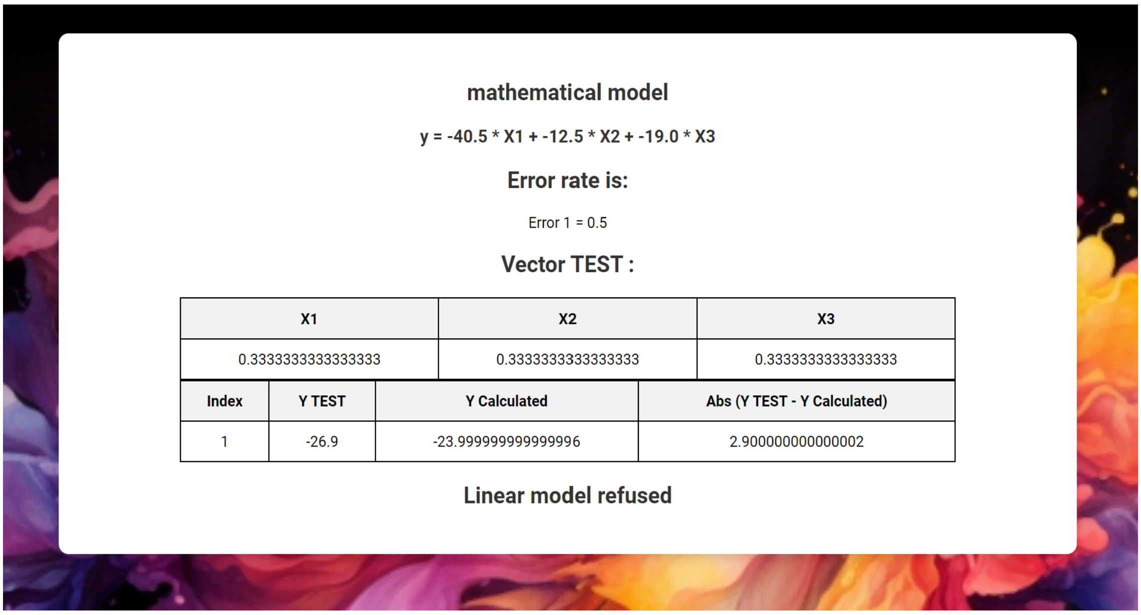
Third, the application display linear model refused



Linear model refused

Figure 4.10: Results Of the linear model

The Results of the first degree of Type 1 summarized in (Figure 4.11).



mathematical model

y = -40.5 * X1 + -12.5 * X2 + -19.0 * X3

Error rate is:

Error 1 = 0.5

Vector TEST :

X1	X2	X3
0.3333333333333333	0.3333333333333333	0.3333333333333333

Index	Y TEST	Y Calculated	Abs (Y TEST - Y Calculated)
1	-26.9	-23.999999999999996	2.900000000000002

Linear model refused

Figure 4.11: Results Of the First-Degree of Type 1

4.7 Example 2 of the Type I (Quadratic)

Consider the example 1.5.1.3.2

4.7.1 Hypothesis of the Quadratic Model

The second-degree model assumes that the response quadratic on the proportions of the components. The mathematical expression of this model is:

$$y = a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3$$

1. Y is the response (cold hold)
2. a_1, a_2,a_3 ,$a_1$2, $a_1$3, and $a_2$3 are the model coefficients
3. X_1, X_2,X_3,X_{12}, X_{13},X_{23} are the proportions of the components

Table 4.3: The tests that need to be performed are shown in the matrix below:

N°	x_1	x_2	x_3	y
1	1	0	0	-40.5
2	0	1	0	-12.5
3	0	0	1	-19
5	1/2	1/2	0	-28.6
6	0	1/2	1/2	-30.8
7	1/2	0	1/2	-18.5

- **Model Equation**

$$y = -40.5x_1 - 12.5x_2 - 19x_3 - 8.4x_1x_2 + 45x_1x_3 - 60.2x_2x_3$$

- **Model Validation** The validity of the first-degree model is tested with a new experiment.

Table 4.4: The matrix of related experiences

N°	x_1	x_2	x_3	Y measure
4	1/3	1/3	1/3	-26.9
8	2/3	1/6	1/6	-29.6
9	1/6	2/3	1/6	-24.2
10	1/6	1/6	2/3	-23.5

- **Observed Response** The observed response are

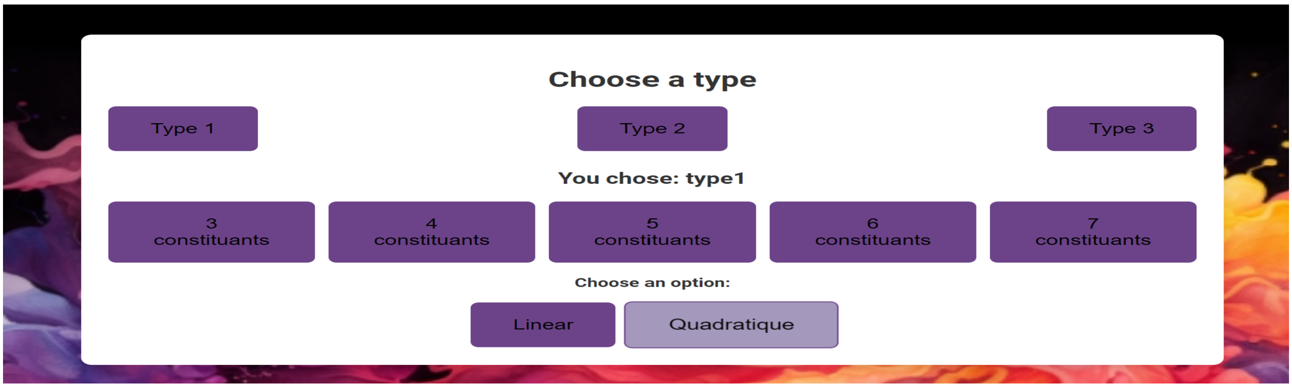
Table 4.5: Y calculated For the model

Y calculated
-26.6
-23.9
-24.6
-23.4

4.7.2 Description of the web application interfaces For TYPE I Quadratic

After choosing a Type (Type I, Type II, Type III) and then selecting the number of constituents (ranging from 3 to 7) for the chosen type. Additionally, the user can choose between two options: Linear or Quadratic option.

We will choose Quadratic Option



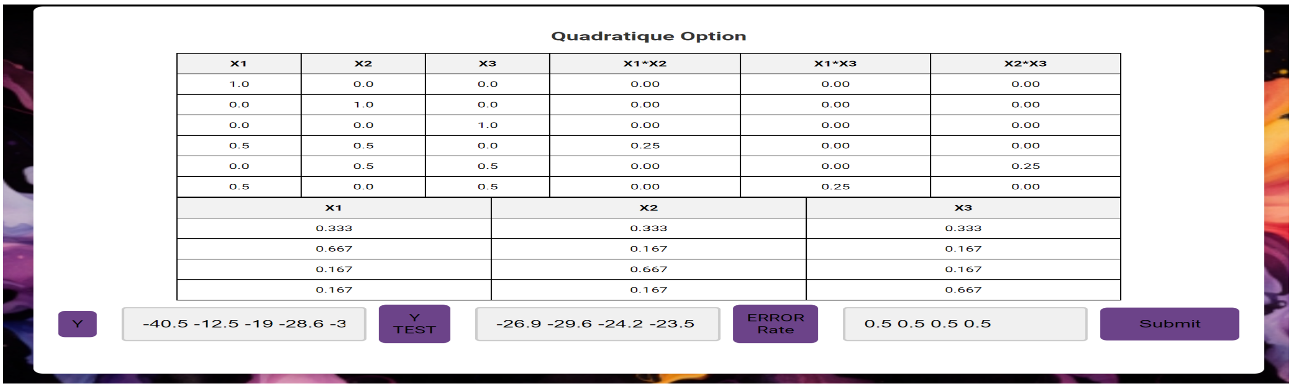
Choose a type

You chose: type1

Choose an option:

Figure 4.12: Chosen a quadratic model

This interface (Figure 4.13) displays the "Quadratic Option," showing a table with columns X_1, X_2, and X_3, each containing binary values and Testing vector values of X1, X2, and X3. Below the table, there are buttons and input fields labeled Y, Y TEST, ERROR Rate, and Submit button, as well as size indicators for the Y and Y TEST inputs.



Quadratique Option

X1	X2	X3	X1*X2	X1*X3	X2*X3
1.0	0.0	0.0	0.00	0.00	0.00
0.0	1.0	0.0	0.00	0.00	0.00
0.0	0.0	1.0	0.00	0.00	0.00
0.5	0.5	0.0	0.25	0.00	0.00
0.0	0.5	0.5	0.00	0.00	0.25
0.5	0.0	0.5	0.00	0.25	0.00

X1	X2	X3
0.333	0.333	0.333
0.667	0.167	0.167
0.167	0.667	0.167
0.167	0.167	0.667

Y: -40.5 -12.5 -19 -28.6 -3

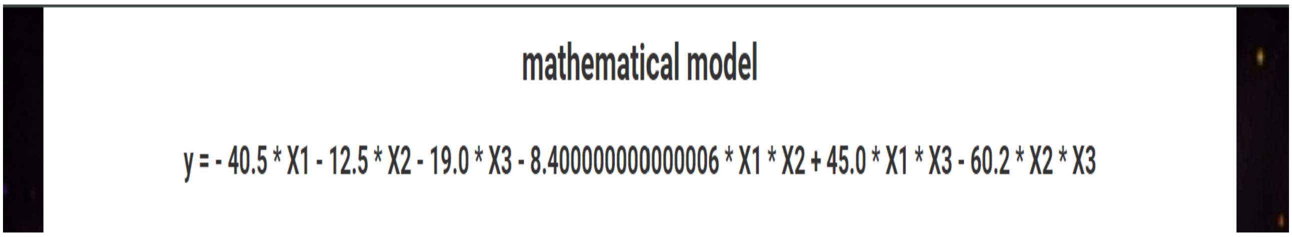
Y TEST: -26.9 -29.6 -24.2 -23.5

ERROR Rate: 0.5 0.5 0.5 0.5

Figure 4.13: Values have been entred for type I

4.7.3 Display Quadratic Results

First,the application display the mathematical model.

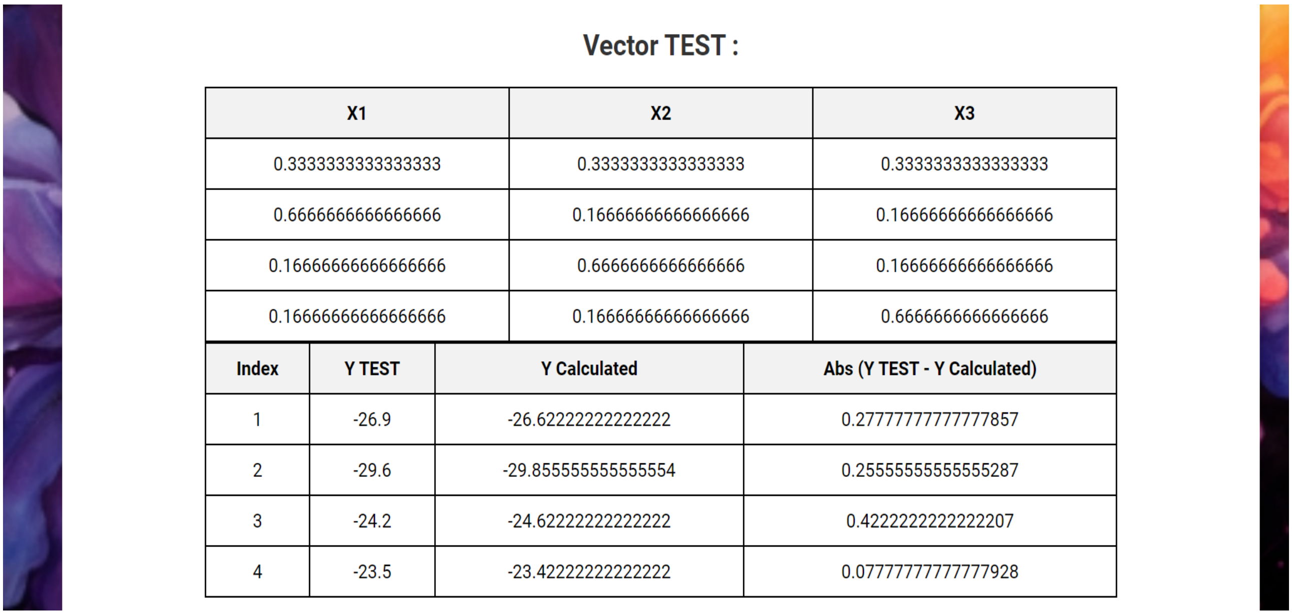


mathematical model

y = - 40.5 * X1 - 12.5 * X2 - 19.0 * X3 - 8.400000000000006 * X1 * X2 + 45.0 * X1 * X3 - 60.2 * X2 * X3

Figure 4.14: Mathematical model

Second ,the application display the difference between y test and y calculated and compare it with the error rate.



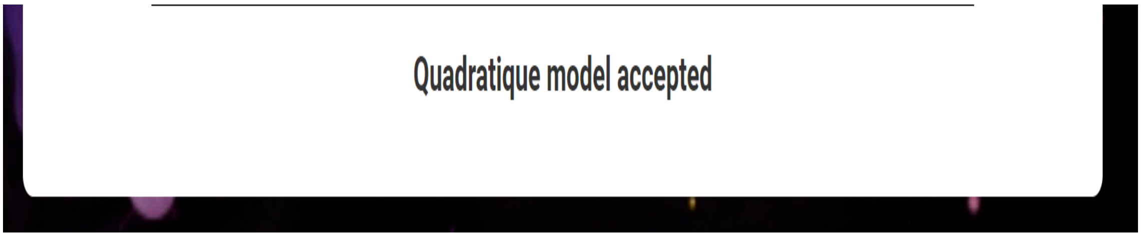
Vector TEST :

X1	X2	X3
0.3333333333333333	0.3333333333333333	0.3333333333333333
0.6666666666666666	0.16666666666666666	0.16666666666666666
0.16666666666666666	0.6666666666666666	0.16666666666666666
0.16666666666666666	0.16666666666666666	0.6666666666666666

Index	Y TEST	Y Calculated	Abs (Y TEST - Y Calculated)
1	-26.9	-26.62222222222222	0.27777777777777857
2	-29.6	-29.855555555555554	0.25555555555555287
3	-24.2	-24.62222222222222	0.4222222222222207
4	-23.5	-23.42222222222222	0.07777777777777928

Figure 4.15: Table displaying results of Y calculated

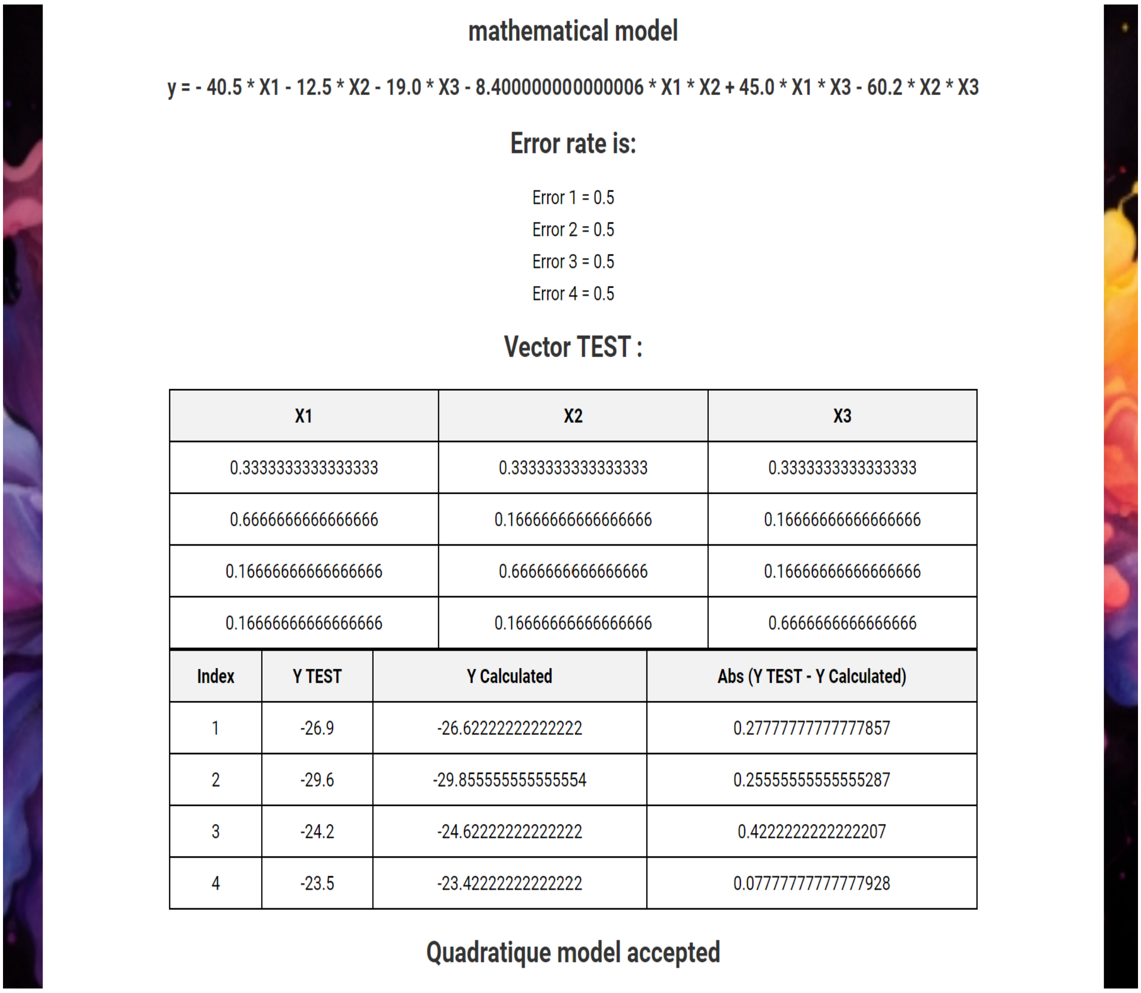
Third, the application display Quadratic model accepted.



Quadratique model accepted

Figure 4.16: Results Of the Quadratic model

The Results of the second degree of Type 1 summarized in (Figure 4.17).



mathematical model

y = - 40.5 * X1 - 12.5 * X2 - 19.0 * X3 - 8.400000000000006 * X1 * X2 + 45.0 * X1 * X3 - 60.2 * X2 * X3

Error rate is:

Error 1 = 0.5

Error 2 = 0.5

Error 3 = 0.5

Error 4 = 0.5

Vector TEST :

X1	X2	X3
0.3333333333333333	0.3333333333333333	0.3333333333333333
0.6666666666666666	0.16666666666666666	0.16666666666666666
0.16666666666666666	0.6666666666666666	0.16666666666666666
0.16666666666666666	0.16666666666666666	0.6666666666666666

Index	Y TEST	Y Calculated	Abs (Y TEST - Y Calculated)
1	-26.9	-26.62222222222222	0.27777777777777857
2	-29.6	-29.855555555555554	0.25555555555555287
3	-24.2	-24.62222222222222	0.4222222222222207
4	-23.5	-23.42222222222222	0.07777777777777928

Quadratique model accepted

Figure 4.17: Results Of the First-Degree of Type 1

4.8 Example 3 of the type III

Consider the example of 1.5.3 of Type III mixture design.

4.8.1 Hypothesis of a Type III mixture design

Table 4.6: Experimental plan to establish the first-degree model and test its validity

Constituent	l_i	U_i	R_i
1	0.25	0.45	0.20
2	0.00	0.20	0.20
3	0.20	0.45	0.25
4	0.00	0.15	0.15

We will passe to our application

First selecting the type III

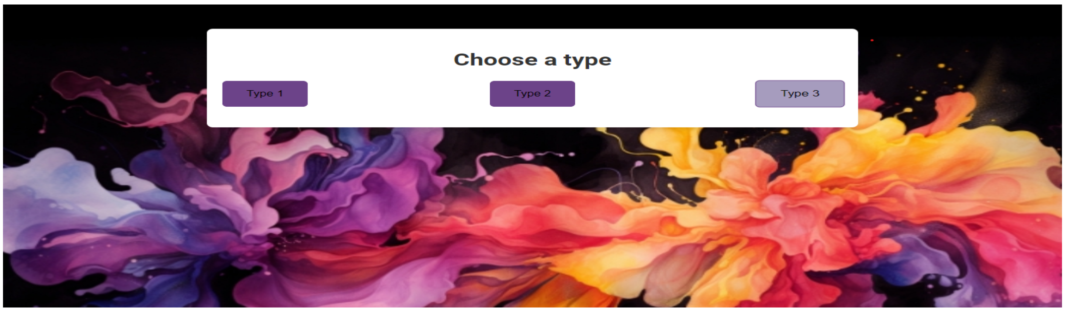
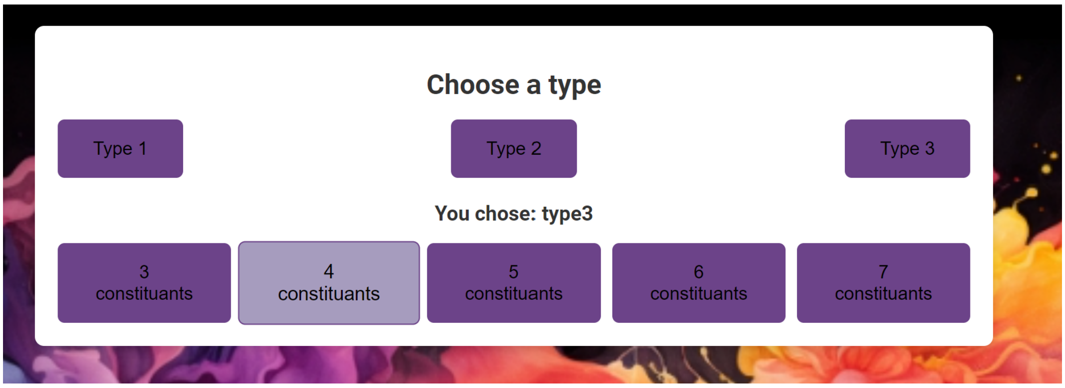


Figure 4.18: Chosen type III

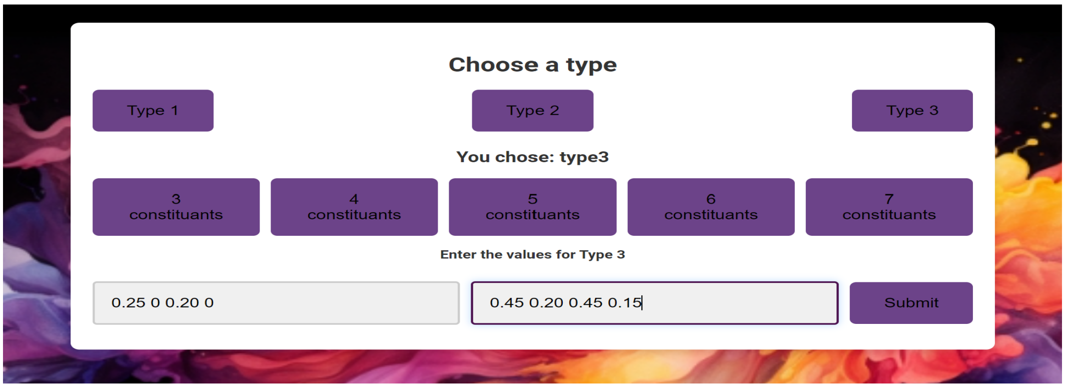
Second Choose four constituent for our four plasticizes (A, B, C and D)



You chose: type3

Figure 4.19: Chosen type III for four constituents

Third ,put the value's lower and upper respecting the dimension then click submit to take it to the display



You chose: type3

Enter the values for Type 3

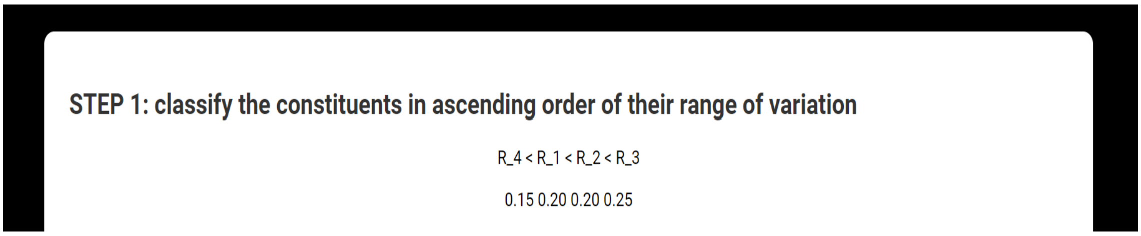
0.25 0 0.20 0

0.45 0.20 0.45 0.15

Figure 4.20: Enter the values for type III

4.8.2 Display

This picture represent 'STEP 1' how to classify the constituents in ascending order



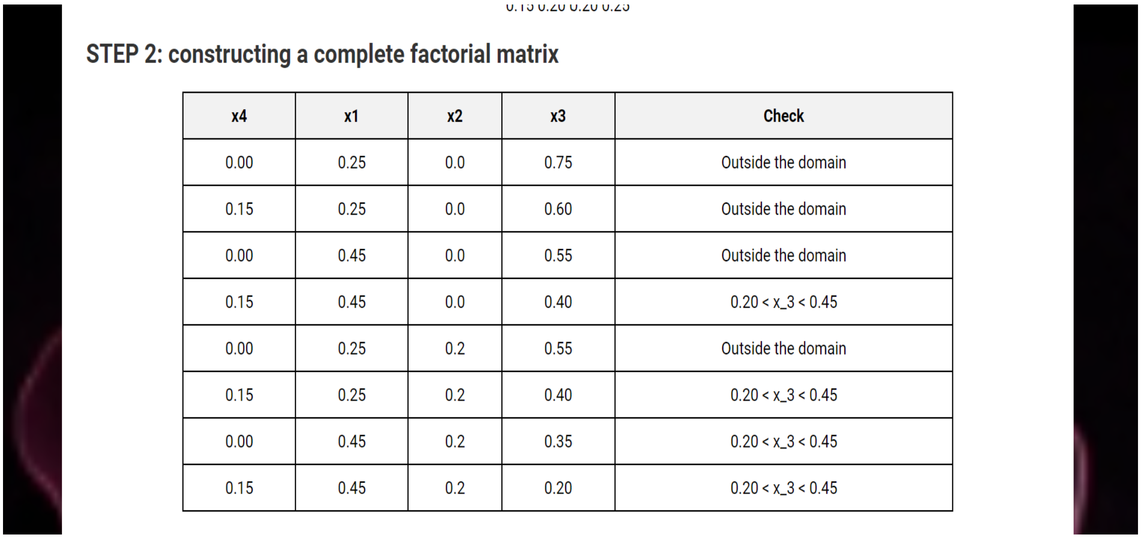
STEP 1: classify the constituents in ascending order of their range of variation

R_4 < R_1 < R_2 < R_3

0.15 0.20 0.20 0.25

Figure 4.21: STEP 1 classify the constituents in ascending order in type III

This picture represent 'STEP 2' how constructing a complete factorial matrix.



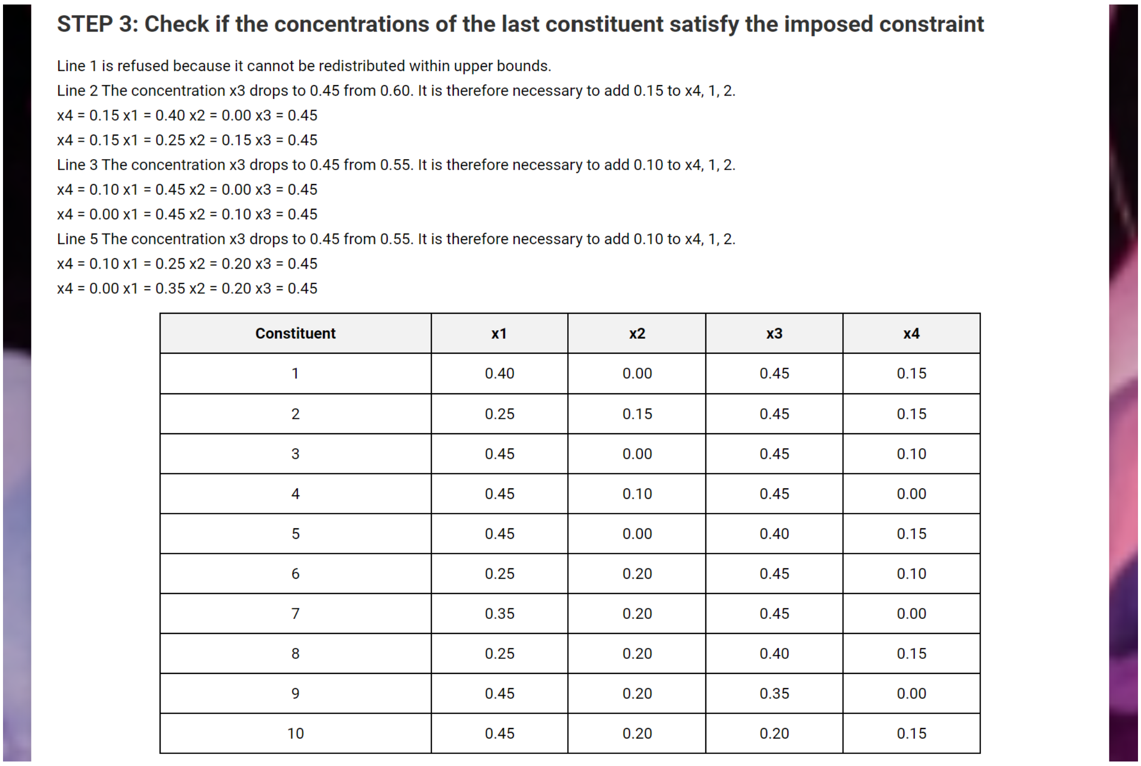
0.15 0.20 0.20 0.25

STEP 2: constructing a complete factorial matrix

x4	x1	x2	x3	Check
0.00	0.25	0.0	0.75	Outside the domain
0.15	0.25	0.0	0.60	Outside the domain
0.00	0.45	0.0	0.55	Outside the domain
0.15	0.45	0.0	0.40	0.20 < x_3 < 0.45
0.00	0.25	0.2	0.55	Outside the domain
0.15	0.25	0.2	0.40	0.20 < x_3 < 0.45
0.00	0.45	0.2	0.35	0.20 < x_3 < 0.45
0.15	0.45	0.2	0.20	0.20 < x_3 < 0.45

Figure 4.22: STEP 2 constructing a complete factorial matrix in type III

This picture represent 'STEP 3' the Check if the concentrations of the last constituent satisfy the imposed constrain



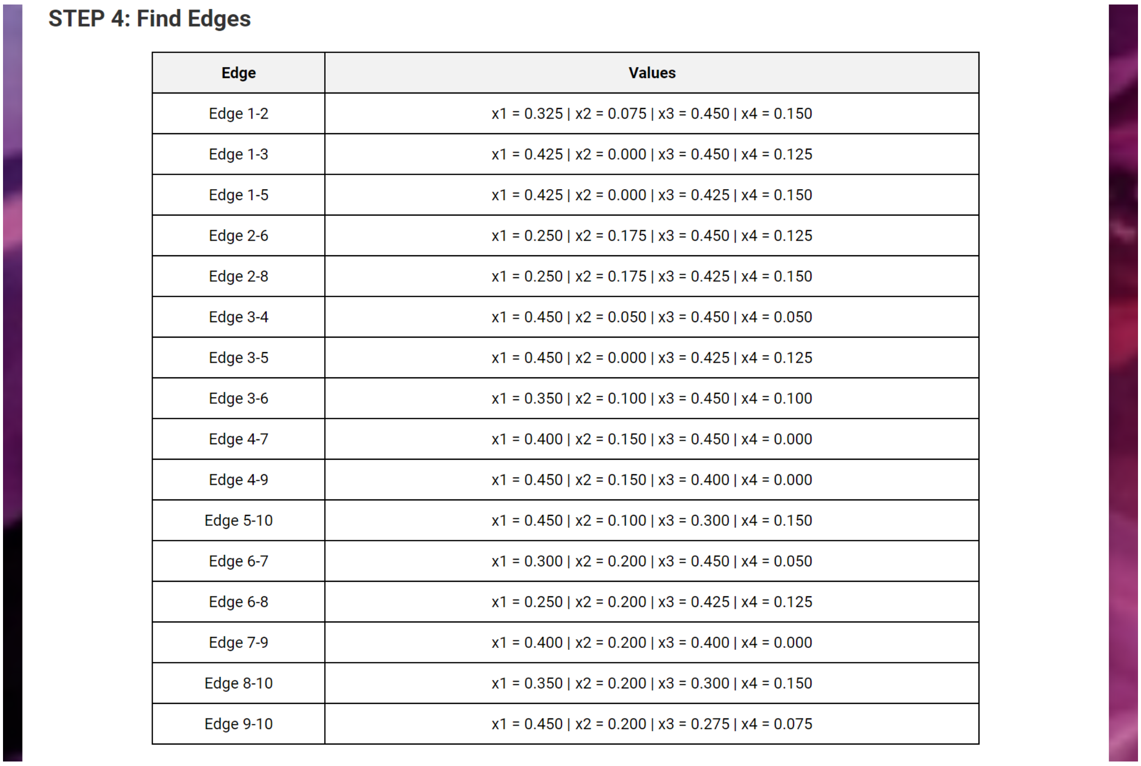
STEP 3: Check if the concentrations of the last constituent satisfy the imposed constraint

Line 1 is refused because it cannot be redistributed within upper bounds.
Line 2 The concentration x3 drops to 0.45 from 0.60. It is therefore necessary to add 0.15 to x4, 1, 2.
x4 = 0.15 x1 = 0.40 x2 = 0.00 x3 = 0.45
x4 = 0.15 x1 = 0.25 x2 = 0.15 x3 = 0.45
Line 3 The concentration x3 drops to 0.45 from 0.55. It is therefore necessary to add 0.10 to x4, 1, 2.
x4 = 0.10 x1 = 0.45 x2 = 0.00 x3 = 0.45
x4 = 0.00 x1 = 0.45 x2 = 0.10 x3 = 0.45
Line 5 The concentration x3 drops to 0.45 from 0.55. It is therefore necessary to add 0.10 to x4, 1, 2.
x4 = 0.10 x1 = 0.25 x2 = 0.20 x3 = 0.45
x4 = 0.00 x1 = 0.35 x2 = 0.20 x3 = 0.45

Constituent	x1	x2	x3	x4
1	0.40	0.00	0.45	0.15
2	0.25	0.15	0.45	0.15
3	0.45	0.00	0.45	0.10
4	0.45	0.10	0.45	0.00
5	0.45	0.00	0.40	0.15
6	0.25	0.20	0.45	0.10
7	0.35	0.20	0.45	0.00
8	0.25	0.20	0.40	0.15
9	0.45	0.20	0.35	0.00
10	0.45	0.20	0.20	0.15

Figure 4.23: STEP 3 verify final component meets requirement.

This picture represent 'STEP 4' find the edges in type III

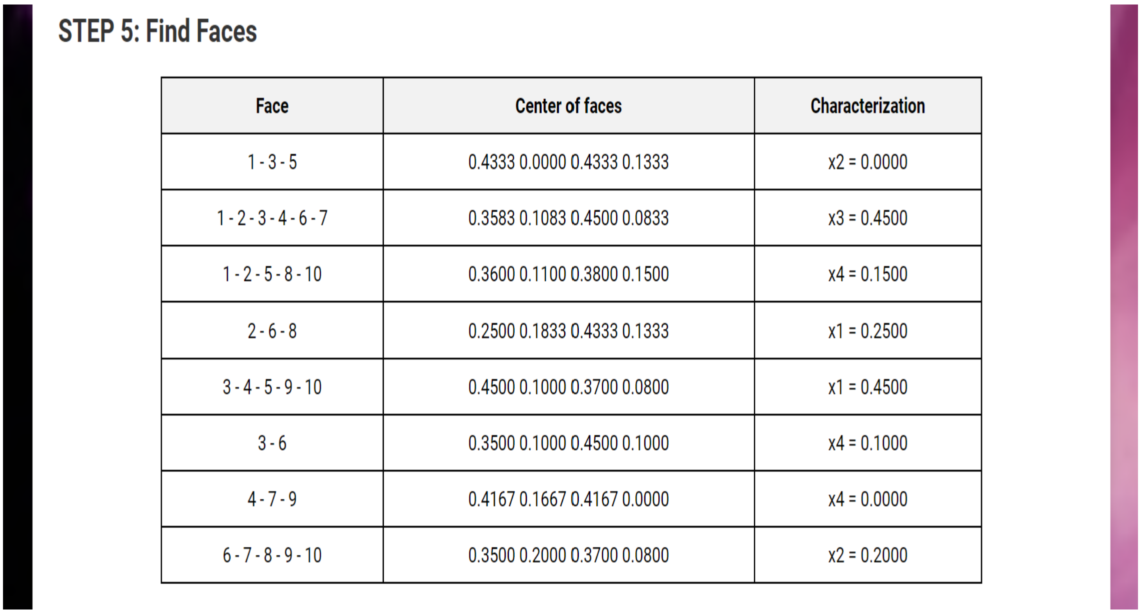


STEP 4: Find Edges

Edge	Values
Edge 1-2	x1 = 0.325 | x2 = 0.075 | x3 = 0.450 | x4 = 0.150
Edge 1-3	x1 = 0.425 | x2 = 0.000 | x3 = 0.450 | x4 = 0.125
Edge 1-5	x1 = 0.425 | x2 = 0.000 | x3 = 0.425 | x4 = 0.150
Edge 2-6	x1 = 0.250 | x2 = 0.175 | x3 = 0.450 | x4 = 0.125
Edge 2-8	x1 = 0.250 | x2 = 0.175 | x3 = 0.425 | x4 = 0.150
Edge 3-4	x1 = 0.450 | x2 = 0.050 | x3 = 0.450 | x4 = 0.050
Edge 3-5	x1 = 0.450 | x2 = 0.000 | x3 = 0.425 | x4 = 0.125
Edge 3-6	x1 = 0.350 | x2 = 0.100 | x3 = 0.450 | x4 = 0.100
Edge 4-7	x1 = 0.400 | x2 = 0.150 | x3 = 0.450 | x4 = 0.000
Edge 4-9	x1 = 0.450 | x2 = 0.150 | x3 = 0.400 | x4 = 0.000
Edge 5-10	x1 = 0.450 | x2 = 0.100 | x3 = 0.300 | x4 = 0.150
Edge 6-7	x1 = 0.300 | x2 = 0.200 | x3 = 0.450 | x4 = 0.050
Edge 6-8	x1 = 0.250 | x2 = 0.200 | x3 = 0.425 | x4 = 0.125
Edge 7-9	x1 = 0.400 | x2 = 0.200 | x3 = 0.400 | x4 = 0.000
Edge 8-10	x1 = 0.350 | x2 = 0.200 | x3 = 0.300 | x4 = 0.150
Edge 9-10	x1 = 0.450 | x2 = 0.200 | x3 = 0.275 | x4 = 0.075

Figure 4.24: STEP 4 find edges

This picture represent 'STEP 5' how we find the faces in type III

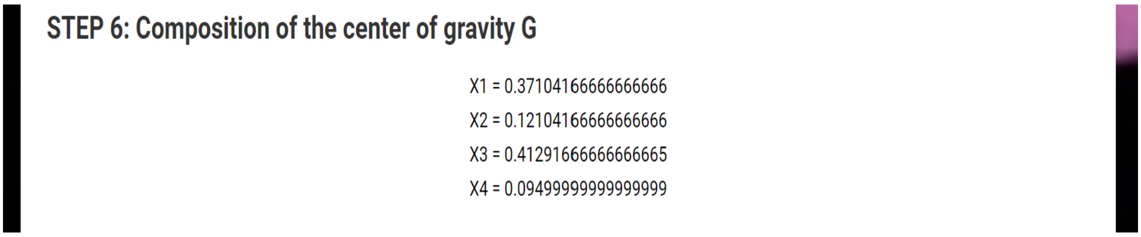


STEP 5: Find Faces

Face	Center of faces	Characterization
1 - 3 - 5	0.4333 0.0000 0.4333 0.1333	x2 = 0.0000
1 - 2 - 3 - 4 - 6 - 7	0.3583 0.1083 0.4500 0.0833	x3 = 0.4500
1 - 2 - 5 - 8 - 10	0.3600 0.1100 0.3800 0.1500	x4 = 0.1500
2 - 6 - 8	0.2500 0.1833 0.4333 0.1333	x1 = 0.2500
3 - 4 - 5 - 9 - 10	0.4500 0.1000 0.3700 0.0800	x1 = 0.4500
3 - 6	0.3500 0.1000 0.4500 0.1000	x4 = 0.1000
4 - 7 - 9	0.4167 0.1667 0.4167 0.0000	x4 = 0.0000
6 - 7 - 8 - 9 - 10	0.3500 0.2000 0.3700 0.0800	x2 = 0.2000

Figure 4.25: STEP 5 find faces

This picture represent 'STEP 6' Composition of the center of gravity G



STEP 6: Composition of the center of gravity G

X1 = 0.37104166666666666
X2 = 0.12104166666666666
X3 = 0.41291666666666665
X4 = 0.09499999999999999

Figure 4.26: STEP 6 Composition of the center of gravity G

This picture represent the resulting graph

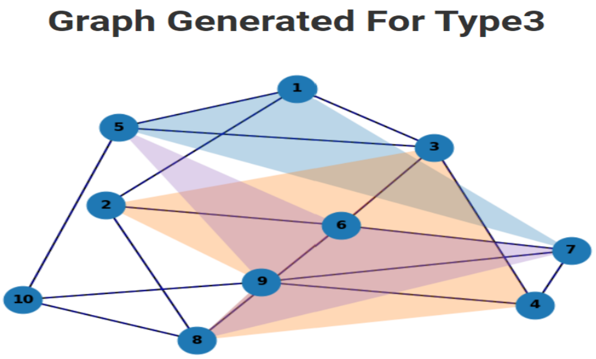
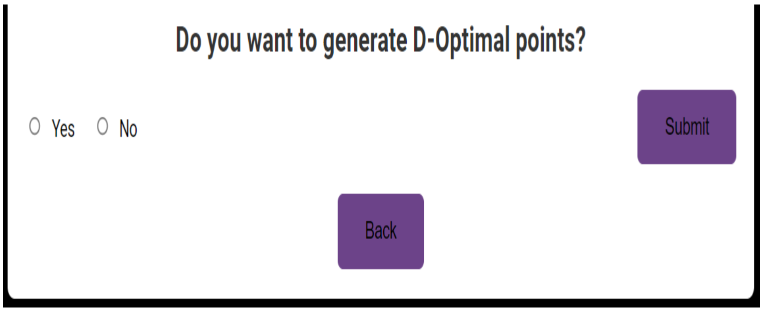


Figure 4.27: The Graph generated using Edges and Faces calculated

4.9 Example 4 Optimality criteria D

After selecting Type 3, the interface displays the following steps: 1, 2, 3, 4, 5, 6. Then, the user is prompted to choose whether they want to calculate the optimal D criteria (figure 4.28).

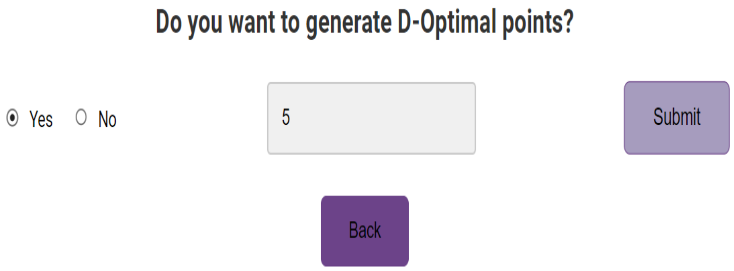


Do you want to generate D-Optimal points?

○ Yes ○ No

Figure 4.28: Selecting D-optimal step

Two radio buttons are presented for this purpose. If the user selects "Yes," an input text field appears where they can enter the value of ’n,’ representing the number of lines (figure 4.29).



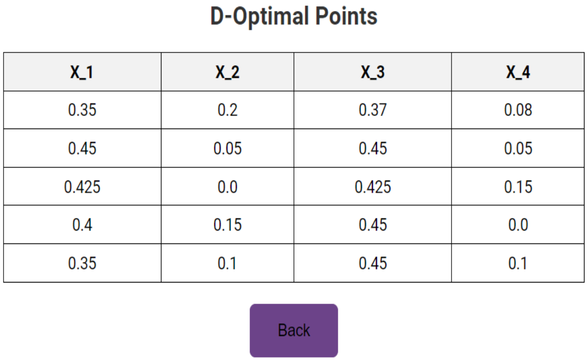
Do you want to generate D-Optimal points?

● Yes ○ No

5

Figure 4.29: chosen and confirming the value of n

The Optimally Criteria D is represented in (Figure 4.30)



D-Optimal Points

X_1	X_2	X_3	X_4
0.35	0.2	0.37	0.08
0.45	0.05	0.45	0.05
0.425	0.0	0.425	0.15
0.4	0.15	0.45	0.0
0.35	0.1	0.45	0.1

Figure 4.30: D-optimal points

4.10 Example 5

We are interested in the cold behavior of a mixture of three constituents. The precision regarding cold resistance is ±0.05.

1. Using a Type I mixture design, specify the steps to execute such a plan, and then provide the study domain.
2. Determine the first-degree model if the responses at the domain vertices are given by: A (1 0 0) → -40.5; B (0 1 0) → -12.5; C (0 0 1) → -19 and the response at the center is -26.9.
3. Study the validity of the model determined in question 2.
4. If the first-degree model is not valid, determine the second-degree model. Points given for this purpose are: D (0.5 0.5 0) → -28.6; E (0 0.5 0.5) → -30.8; F (0.5 0 0.5) → -18.5.
5. Study the validation of the model using the points: G (0.667 0.167 0.167) → -29.6; H (0.167 0.667 0.167) → -24.2; I (0.167 0.167 0.667) → -23.5.

Provide the study domain, specifying the location of all points A, B, C, D, E, F, G, H, and I.

4.10.1 Solution

1. The approach to use is:
2. Assume a first-degree model and perform mixtures according to the Scheffé lattice k,1, then calculate the model coefficients.
3. Test the validation of the model by performing one or more mixtures within the domain. If validation is accepted, the problem is resolved.
4. If validation is not accepted, assume a second-degree model and perform mixtures to complete the Scheffé lattice $k,2$, then calculate the model coefficients. Validation is tested as above

Study Domain

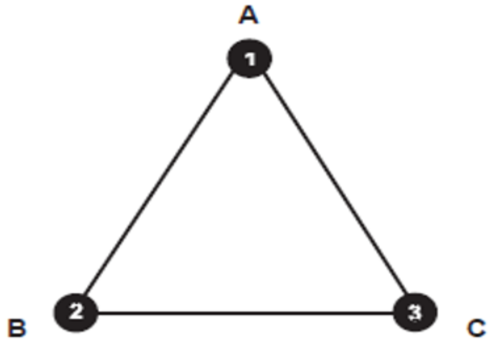


Figure 4.31: Study domain

The model is written :

$$Y = b_1X_1 + b_2X_2 + b_3X_3$$

Table 4.7: The tests that need to be performed are shown in the matrix below:

Point	x$_1$	x$_2$	x$_3$	cold storage
A	1	0	0	-40.5
B	0	1	0	-12.5
C	0	0	1	-19
D	1/3	1/3	1/3	-26.9

The three points A, B, and C allow us to write a system of 3 equations with 3 unknowns:

$$-40.5 = b_1 * 1 + b_2 * 0 + b_3 * 0$$

$$-12.5 = b_1 * 0 + b_2 * 1 + b_3 * 0$$

$$-19.0 = b_1 * 0 + b_2 * 0 + b_3 * 1$$

Obviously, the model is written :

$$Y = -40.5X_1 + -12.5X_2 + -19.0X_3$$

Model validation

1. For the calculated value of cold resistance : y=-24
2. For the observed value of cold resistance: y=-26.9

The difference between the observed and calculated values is significantly greater than the precision of 0.05. Therefore, the first-degree model is rejected

Hypothesis of the second-degree model

The model is written as::

$$y = a_1x_1 + a_2x_2 + a_3x_3 + a_{12}x_1x_2 + a_{13}x_1x_3 + a_{23}x_2x_3$$

The Scheffé network is composed of points A, B, C, E, F, and G. The table below represents the complementary trials:

Table 4.8: The tests that need to be performed are shown in the matrix below:

Point	x_1	x_2	x_3	cold storage
E	0.5	0.5	0	-28.6
F	0.5	0	0.5	-3.8
G	0	0.5	0.5	-18.5

These 6 trials allow us to write a system of 6 equations.:

$$a_1 = -40.5$$

$$a_2 = -12.5$$

$$a_3 = -19$$

$$a_12 = 4a_1 - 5a_2 - 2a_3 = -8.4$$

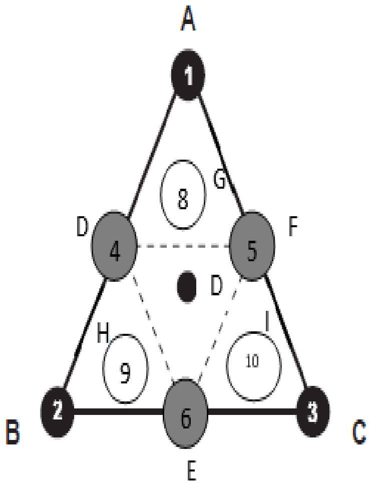
$$a_13 = 4a_1 - 2a_2 - 2a_3 = 45$$

$$a_23 = 4a_1 - 2a_2 - 2a_3 = -60.2$$

Table 4.9: The tests that need to be performed are shown in the matrix below:

Point	x_1	x_2	x_3	Observed cold resistance	Calculated cold resistance	Difference
G	2/3	1/6	1/6	-29.6	-26.9	0.3
H	1/6	2/3	1/6	-24.2	-29.9	0.4
I	1/6	1/6	2/3	-23.5	-24.6	0.1

With a precision of 0.5, the model is accepted.



4.11 Example 6

We are studying a mixture of four components whose proportions are constrained by the following limits expressed as percentages:

$$10 \leq x_1 \leq 90$$

$$10 \leq x_2 \leq 50$$

$$10 \leq x_3 \leq 30$$

$$0 \leq x_4 \leq 5$$

1. Show the existence of the experimental domain.
2. Provide the coherent experimental domain.
3. By applying the Snee and Marquardt algorithm, determine:
   (a) The vertices of the experimental domain.
   (b) The edges and the composition of their midpoint.
   (c) The faces and the composition of their center.
   (d) The composition of the centroid of the experimental domain.

4.11.1 Solution

1. To ensure the experimental domain can exist, it is sufficient that the conditions $L = \sum li < 100$ and $U = \sum ui > 100$ both conditions are satisfied (where ℓ_i and u_i denote

the lower and upper limits of constituent i).

If either of the two previous conditions is not satisfied, then the domain does not exist. We have **L**= 10+10+10+0=30 <**100** and U=90+50+30+5=175>**100**. With both conditions verified, the experimental domain exists..

2. The proposed limits of the mixture components and the corresponding domain d_i are represented in the following table:

Table 4.10: The tests that need to be performed are shown in the matrix below:

Constituent	l_i%	U_i%	D_i%
A	10	90	80
B	10	50	40
C	10	30	20
D	0	5	5

Actual upper limit

We should have

$$d_i \leq 100 - L$$

If the concentration domain d_i of a constituent exceeds the quantity 100-L=70, its upper limit must be adjusted. For constituent A, we have $d_A = 80$ which exceeds 70 by 10. The upper limit of constituent A must be reduced accordingly and becomes 90-10=80.

Actual lower limit

We should have

$$d_i \leq U - 100$$

Every time the range of a constituent exceeds the quantity 100-U=75, its lower limit must be adjusted. For constituent A, we have dA=80>75. dA exceeds the quantity 100-U by 80-75=5.The lower limit must be increased by this amount, thus becoming 10+5=15.

Coherent experimental domain:

$$15 \leq x_1 \leq 80$$

$$10 \leq x_2 \leq 50$$

$$10 \leq x_3 \leq 30$$

$$0 \leq x_4 \leq 5$$

3. Snee and Marquardt Algorithm

a) Vertices of the experimental domain

1. The $k = 4$ constituents A,B,C and D with concentrations $x_1, x_2, x_3 and x_4*$ respectively , are arranged in increasing order of their domains, namely:

D , C , B , A

2. We construct with the $k - 1 = 4 - 1 = 3$ first constituents, a 2^3 full factorial design. where li corresponds to the level -1 and ui to the level +1. We fill the column of the last constituent in a way that respects the general constraint of mixtures $\sum x_i = 100$. Applying this to our example leads to the following table:

Mix number	D	C	B	A
1	0	10	10	80
2	5	10	10	75
3	0	30	10	60
4	5	30	10	55
5	0	10	50	40
6	5	10	50	35
7	0	30	50	20
8	5	30	50	15

For the coherent domain, we have L=35 et U=165 Therefore :

$$100 - L = 65$$

$$U - 100 = 65$$

3. All constraints of constituent A satisfy the constraints of the coherent domain or actual domain. No constraints of constituent A need to be modified. Therefore, the table below represents the 8 vertices of the experimental domain.

Mix number	A	B	C	D
1	80	10	10	0
2	75	10	10	5
3	60	10	30	0
4	55	10	30	5
5	40	50	10	0
6	35	50	10	5
7	20	50	30	0
8	15	50	30	5

b) Edges and the composition of their midpoint

After identifying the vertices, we need to identify the edges, faces, hyper faces, and other boundaries of the domain. Depending on the number k of constituents, the domain is bounded by:

4. Edges($k = 3$)

5. Edges and faces($k = 4$)

6. ...

- edges, faces, ... hyper faces of dimension $r = k - 2$

All mixtures on a boundary of dimension r share the concentrations of p constituents, where p is determined by the relationship:

$$p = k - r - 1$$

In our case $k = 4$, hence $r = 4 - 2 = 2$ Therefore, the domain is bounded only by edges and 2 dimensional faces. he search for the boundaries of the domain is conducted based on the table of vertices (table 2).

To search for the edges, we take $r = 1$ hence $p = k - r - 1 = 4 - 1 - 1 = 2$

7. Edges starting from vertex 1. Vertex 1 has the composition:

$$A = 80, B = 10, C = 10, D = 0$$

We find the values:

$$B = 10 \quad and \quad C = 10 \quad on \quad line \quad 2 \quad of \quad table \quad 2$$

$$B = 10 \quad et \quad D = 0 \quad on \quad line \quad 3$$

$$C = 10 \quad D = 0 \quad on \quad line \quad 5$$

We have identified the 2 edges **1-2,1-3, 1-5**

The midpoint of edge 1-2 and 1-3 respectively has the composition:

$$A = (80 + 75)/2 = 77,5 \ B = 10 \ C = 10 \ D = 2,5$$

$$A = (80 = 60)/2 = 70 \ B = 10 \ C = 20 \ D = 0$$

8. Edges starting from vertex 2. Vertex 2 has the composition

$$A = 75,\ B = 10,\ C = 10,\ D = 5$$

We find the values:

$$B = 10,\ D = 5, \quad on \quad line \quad 4$$

We have identified edge **2-4**.

We have identified the 2 edges from vertex 3. Vertex 3 has the composition:

$$A = 60,\ B = 10,\ C = 30,\ D = 0$$

We find the values:

$$B = 10 \quad C = 30 \quad on\ line \quad 4$$

$$C = 30 \quad D = 0 \quad on\ line \quad 7$$

We have identified the 2 edges **3-4, 3-7**

9. Edges starting from vertex 4. Vertex 4 has the composition

$$A = 55,\ B = 10,\ C = 30,\ D = 5$$

We find the values:

$$C = 30 \quad D = 5 \quad on \quad line \quad 8$$

We have identified edge **4-8**

10. Edges starting from vertex 5. Vertex 4 has the composition

$$A = 40,\ B = 50,\ C = 10,\ D = 0$$

We find the values:

$$B = 50 \quad C = 10 \quad on \quad line \quad 6$$

$$B = 50 \quad D = 0 \quad on \quad line \quad 7$$

We have identified edge **5-6, 5-7**

11. Edges starting from vertex 6. Vertex 6 has the composition

$$A = 35, \; B = 50, \; C = 10, \; D = 5$$

We find the values:

$$B = 50 \quad D = 5 \quad on \quad line \quad 8$$

We have identified edge **6-8**

12. Edges starting from vertex 7. Vertex 7 has

$$A = 20, \; B = 50, \; C = 30, \; D = 0$$

We find the values:

$$B = 50 \quad C = 30 \quad on \quad line \quad 8$$

We have identified edge **7-8**

Domain Edges	A	B	C	D
1	80	10	10	0
2	75	10	10	5
3	60	10	30	0
4	55	10	30	5
5	40	50	10	0
6	35	50	10	5
7	20	50	30	0
8	15	50	30	5

c) Faces and the composition of their center

With the values $k = 4$, $r = 2$ then $p = 4 - 2 - 1 = 1$

13. Faces containing vertex 1:

$$A = 80,\ B = 10,\ C = 10,\ D = 0$$

Vertices 2, 3, and 4 define with vertex 1 a face characterized by B=10, and its center has the composition:

$$A = (80 + 75 + 60 + 55)/4 = 67,5,\ B = 10,\ C = (10 + 10 + 30 + 30)/4 = 20,\ D = 2,5$$

The face 1-2-5-6 is characterized by C=10.
The face 1-3-5-7 is characterized by D=0.

14. Faces containing vertex 2

We find the faces:

$$2 - 3 - 4 \textit{ characterized by } B = 10$$

$$2 - 5 - 6 \textit{ characterized by } C = 10$$

2-4-6-8 characterized by D=5

15. Faces containing vertex 3

We find the faces:

$$3 - 4 - 7 - 8 \textit{ characterized by} = 30$$

$$3 - 5 - 7 \textit{ characterized by } D = 0$$

16. Faces containing vertex 4

We find the faces:

$$4 - 7 - 8 \textit{ characterized by } C = 30$$

$$4 - 6 - 8 \textit{ characterized by } D = 5$$

17. Face containing vertex 5

We find the face:

$$5-6-7-8 \textit{ characterized by } B=50$$

18. Face containing vertex 6:

We find the face:

$$6-7-8 \textit{ characterized by } B=50$$

Some faces are repeated two or more times.
For example, faces 1-2-5-6 and 2-5-6, both characterized by C=10, the first counted from vertex 1 and the second from vertex 2. Ultimately, all distinct faces are represented in the following table:

Domain Faces	A	B	C	D	Caracter
1-2-3-4	67,5	10	20	2,5	B=10
1-2-5-6	57,5	30	10	2,5	C=10
1-3-5-7	53,75	30	20	0	D=0
2-4-6-8	45	30	20	5	D=5
3-4-7-8	37,5	30	30	2,5	C=30
5-6-7-8	27,5	50	20	2,5	B=50

4.12 Example 7

To model (establishing a quadratic model) and simulate the behavior of a system (a product with 3 constituents), the following set of individual constraints is provided:

$$0,2 \leq x_1 \leq 0,6$$

$$0,1 \leq x_2 \leq 0,6$$

$$0,1 \leq x_3 \leq 0,5$$

1. Study the existence of the experimental domain
   Verify the consistency of the individual constraints. To do this, we will compare the range $R_i = b_i - a_i$ of each constituent i (where a_i and b_i denote respectively the lower constraint and the upper constraint of the constituent i) to $R_a = 1 - \sum_{i=1}^{3} a_i$, the linear measurement of simplex A, and $R_b = \sum_{i=1}^{3} b_i - 1$, the linear measurement of simplex B.

2. Specify the coherent experimental domain.

3. By applying the Snee and Marquardt algorithm, determine the composition of each vertex.

4. Provide all edges and the composition of their midpoint.

5. Give the composition of the center of gravity of the study domain.

6. What is the set C of experimental points that can be candidates for establishing mathematical models (linear, quadratic, or cubic models)?

4.12.1 Solution

$$0,2 \leq x_1 \leq 0,6$$

$$0,1 \leq x_2 \leq 0,6$$

$$0,1 \leq x_3 \leq 0,5$$

1.Existence of the experimental domain

$$L = \sum_{i=1}^{3} a_i = 0.2 + 0.1 + 0.1 = 0.4 < 1$$

$$U = \sum_{i=1}^{3} b_i = 0.6 + 0.6 + 0.5 = 1.7 > 1$$

Since L<1 and U>1 the experimental domain exists.

2. Verification of the consistency of individual

constraints Ranges of constituents (or concentration domains of constituents):

$$R_1 = 0.6 - 0.2 = 0.4$$

$$R_2 = 0.6 - 0.1 = 0,5$$

$$R_3 = 0.5 - 0.1 = 0.4$$

$$R_a = 1 - L = 1 - 0.4 = 0.6$$

$$R_b = U - 1 = 1.7 - 1 = 0.7$$

1. If the range R_i exceeds the quantity R_a, then the upper limit of the corresponding constituent is incompatible and must be corrected and replaced by the new upper constraint, known as the implicit constraint:

$$b_i = a_i + R_a.$$

In our case, no upper limit needs to be corrected since all R_i are less than R_a

2. The range R_i exceeds the quantity R_b ., If the range exceeds the quantity, then the lower limit of the corresponding constituent is incompatible and must be corrected and replaced by the new implicit lower constraint:

$$a_i = b_i + R_b$$

In our case, no lower limit needs to be corrected since all R_i are less than R_b

3. Coherent experimental domain

The coherent experimental domain and the corresponding implicit ranges are therefore:

$$0.2 \leq x_1 \leq 0.6 \quad R_1 = 0.4$$

$$0.1 \leq x_2 \leq 0.6 \quad R_2 = 0.5$$

$$0.1 \leq x_3 \leq 0.5 \quad R_3 = 0.4$$

The measurements of simplex A and B remain the same.: $R_a = 0.6$ $R_b = 0.7$

4. Composition of the vertices of the polygon

Constituent	a$_i$	b$_i$	r$_i$
1	0.2	0.6	0.4
2	0.1	0.6	0.5
3	0.1	0.5	0.4

The experimental domain formed by the set of coherent constraints has 6 vertices. To determine the composition x_1, x_2, x_3 To determine the composition of each vertex of the convex polygon forming the experimental domain, we apply the **Snee and Marquardt** (or MacLean and Anderson) algorithm. The different steps of this method are as follows:

Step 1: The q=3 constituents are classified in increasing order of their domains

$$R1 = R3 = 0.4 \quad R2 = 0.5$$

$$R_1 \quad R_3 \quad R_2$$

Step 2: Construct with the q-1=2 first constituents a 2-level full factorial design. Corresponding to level -1 and to level +1. Fill the column of the last constituent in a way that respects the general constraint of mixtures $x_i = 1$. Applying this to our example leads to the following table :

Constituent	x_1	x_2	x_3	$0.1 \leq x_2 \leq 0.6$
1	0.2	0.1	0.7	out of domain
2	0.6	0.1	0.3	
3	0.2	0.5	0.3	
4	0.6	0.5	0.1	out of domain

1. Step 3: Verify if the concentrations of the last constituent satisfy the imposed constraints. If yes, the corresponding row is a vertex of the domain. If not, adjust the last constituent to its nearest limit. Then, readjust the sum of concentrations to 1 by modifying one of the concentrations of the q-1 first constituents. All possible solutions correspond to new vertices Lines 2 and 3 For rows 2 and 3, we find the vertices:

$$x1 = 0.6, x3 = 0.1, x2 = 0.3$$

$$x1 = 0.2, x3 = 0.5, x2 = 0.3$$

Line 1

On the first row of the table, the concentration x_2 is brought back from 0.7 to 0.6 . Thus 0.1 is to be added either to x_1 or to x_3 . This leads us to the two vertices :

$$x1 = 0.3, x3 = 0.1, x2 = 0.6$$

$$x1 = 0.2, x3 = 0.2, x2 = 0.6$$

Line 4

On line 4, the concentration x_2 is adjusted to 0.1 . Therefore 0.2 can be subtracted either from x_1 or x_3. This leads us to the two vertices:

$$x1 = 0.4, x3 = 0.5, x2 = 0.1$$

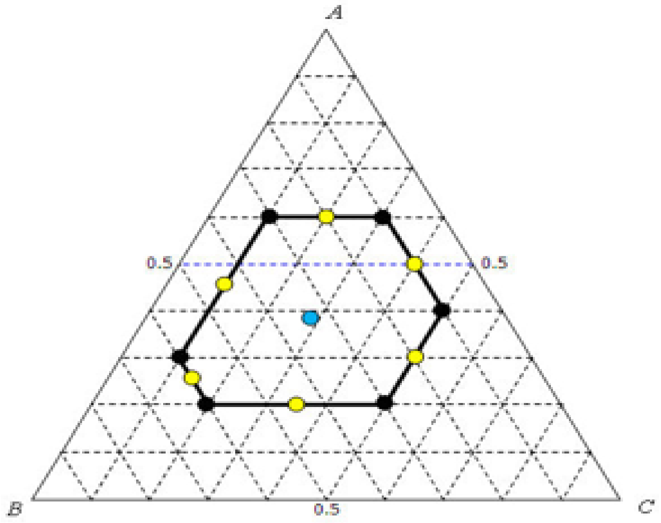
$$x1 = 0.6, x3 = 0.3, x2 = 0.1$$

Vertex	x_1	x_2	x_3
1	0.6	0.3	0.1
2	0.2	0.3	0.5
3	0.3	0.6	0.1
4	0.2	0.6	0.2
5	0.4	0.1	0.5
6	0.6	0.1	0.3

By arranging the vertices in a clockwise direction, we obtain the following final table (see figure):

Composition of the 6 vertices

Vertex	x$_1$	x$_2$	x$_3$
1	0.6	0.3	0.1
2	0.2	0.1	0.3
3	0.4	0.1	0.5
4	0.2	0.3	0.5
5	0.2	0.6	0.2
6	0.3	0.6	0.1



Polygon edges and the composition of their midpoint

All mixtures on a boundary of dimension share the concentrations of p constituents, p being given by the relation:

$$p = q - r - 1$$

Thus, the mixtures located on the same edge ($r = 1$) have $p = 3 - 1 - 1 = 1$ common concentrations. The search for edges in the domain is done starting from vertex 1, where we look in the rest of the table for vertices that share with it $p = 1$ common values. We repeat the process with vertices 2, 3, 4, and 5. After identifying the vertices of an edge, we calculate the compositions of their midpoint by averaging the concentrations of the corresponding vertices.

Edges involving vertex 1

The vertex 1 has the composition:

$$x_1 = 0.6, \quad x_2 = 0.3, \quad x_3 = 0.1$$

We find the values $x_1 = 0.6$ on line 2 and $x_3 = 0.1$ on line 6. We have identified two edges of the domain starting from vertex 1: the edges $1 - 2$ and $1 - 6$. The midpoint of each of these two edges has the following composition, respectively:

$$\begin{array}{lll} x_1 = 0.6 & x_2 = 0.2 & x_3 = 0.2 \\ x_1 = 0.45 & x_2 = 0.45 & x_3 = 0.1 \end{array}$$

Edge starting from vertex 2

The vertex 2 has the composition:

$$x_1 = 0.6, \quad x_2 = 0.1, \quad x_3 = 0.3$$

We find the values $x_1 = 0.1$ on the line 3. We have identified the edge $2 - 3$ whose midpoint has the composition:

$$x_1 = 0.5, x_2 = 0.1, x_3 = 0.4$$

Edge starting from vertex 3

The vertex 3 has the composition:

$$x_1 = 0.4, \quad x_2 = 0.1, \quad x_3 = 0.5$$

We find the values $x_3 = 0.5$ on the line 4. We have identified the edge $3 - 4$ whose midpoint has the composition:

$$x_1 = 0.3, x_2 = 0.2, x_3 = 0.5$$

Edge starting from vertex 4

The vertex 4 has the composition:

$$x_1 = 0.2, \quad x_2 = 0.3, \quad x_3 = 0.5$$

We find the values $x_1 = 0.2$ on the line 5. We have identified the edge $4 - 5$ whose midpoint has the composition:

$$x_1 = 0.2, x_2 = 0.45, x_3 = 0.35$$

Edge starting from vertex 5

The vertex 5 has the composition:

$$x_1 = 0.2, \quad x_2 = 0.6, \quad x_3 = 0.2$$

We find the values $x_2 = 0.6$ on the line 6. We have identified the edge $5 - 6$ whose midpoint has the composition:

$$x_1 = 0.25, x_2 = 0.6, x_3 = 0.15$$

In total, the domain has 6 edges listed in the following table:

Edges and the composition of their midpoint

Edge	x_1	x_2	x_3
1-2	$x_1 = 0.6$	$x_2 = 0.2$	$x_3 = 0.2$
1-6	$x_1 = 0.45$	$x_2 = 0.45$	$x_3 = 0.1$
2-3	$x_1 = 0.5$	$x_2 = 0.1$	$x_3 = 0.4$
3-4	$x_1 = 0.3$	$x_2 = 0.2$	$x_3 = 0.5$
4-5	$x_1 = 0.2$	$x_2 = 0.45$	$x_3 = 0.35$
5-6	$x_1 = 0.25$	$x_2 = 0.6$	$x_3 = 0.15$

Center of Gravity Composition of the Study Domain

The composition of the center of gravity G is obtained by calculating the average of the compositions of the 6 vertices. The result is:

$$x_1 = 0.3833, \quad x_2 = 0.3333, \quad x_3 = 0.2833$$

Set C of candidate points

The introduction of double constraints generally modifies the shape of the experimental domain. We observe that the shape of the domain and the number of vertices depend on the limits l_i and u_i. Scheffé networks are no longer usable. The approach will therefore consist of:

- Identifying the vertices, then the edges, faces, ..., in general, the boundaries of the domain, knowing that it is always convex in shape.
- Creating a set C of candidate points likely to appear in the design. By analogy with what we have seen for Scheffé networks, these candidate points are vertices, midpoints of edges, centers of faces, ... The choice depends on the degree of the model being sought.
- Selecting from C candidates the n points in the design based on the criterion of maximum determinant or an equivalent criterion.

For our example, set C consists of vertices, midpoints of edges, and the barycenter (or center of gravity), which constitutes the center of the only face of the domain since we have a ternary system. Set C is presented in the following table:

Set C of Candidate Points

	N°	x_1	x_2	x_3
Mixtures	1	0.6	0.3	0.1
for	2	0.6	0.1	0.3
the	3	0.4	0.1	0.5
first	4	0.2	0.3	0.5
degree	5	0.2	0.6	0.2
	6	0.3	0.6	0.1
Mixtures	7	0.6	0.2	0.2
for	8	0.45	0.45	0.1
model	9	0.5	0.1	0.4
quadratic	10	0.3	0.2	0.5
	11	0.2	0.45	0.35
	12	0.25	0.6	0.15
centroid	13	0.3833	0.3333	0.2833

This selection allows:

- Calculating the first-degree model using 6 vertices and using the other 6 mixtures and the centroid to test the validity of this model.
- Using the remaining mixtures to estimate, if necessary, the coefficients of the second-degree model, with the centroid still serving as a test point.

Conclusion

The Methodology of Experimental Research, also known as experimental design, is a mathematical discipline within inferential statistics. The theory of experimental design methods is still evolving and has gained significant importance due to the availability of specialized software. The work proposed to us had two main objectives:

Firstly, to study mixture experimental designs, which are widely used in the pharmaceutical industry as well as in many other fields. Secondly, to develop our own application. This will allow us not only to avoid tedious calculations but also to process and analyze any practical experiment in chemical, pharmaceutical, agronomic, physical domains, etc. Developing specific software for mixture designs would be very interesting.

Our research has shown that the use of experimental design methodology is crucial for optimizing experimental processes and ensuring reliable and interpret able results. By using geometric and matrix representations, we have provided a comprehensive overview of the main types of mixture designs: Type I Mixture Designs, Type II Mixture Designs, and Type III Mixture Designs. Additionally, we have developed a by using python software application to facilitate the construction and analysis of these designs, providing a practical tool for validating mathematical models.

the Methodology of Experimental Research plays an indispensable role in various scientific and industrial fields. By focusing on mixture experimental designs, we have addressed a critical need in industries such as pharmaceuticals, chemicals, agronomy, and beyond. Our work demonstrates not only the theoretical importance of these designs but also offers practical solutions through the development of an software application.

In the future, the creation of specialized software for mixture designs could further streamline and enhance the experimental process across multiple sectors. The oil and natural gas industry, in particular, stands to benefit significantly from such advancements, given the complexity and scale of their experimental needs. The ongoing development of experimental design theory and its practical applications holds great promise for future innovations and efficiencies in experimental research.

Bibliography

[Mitchell (1974)] T. J. Mitchell : An Algorithm for the Construction of D-Optimal Experimental Designs. Technometrics, 16(2):203-210, 1974.

[Montgomery (1991)] D. C. Montgomery : Design and Analysis of Experiments, third edn, John Wiley and Sons, INC., USA, 1991.

[Umetrics (2006)] Umetrics : MODDE 8 - User Guide Tutorial, 8.0 edn, Umeå, 2006.

[Wu and Hamada (2000)] C. F. J. Wu and M. Hamada : Design of Experiments: Planning, Analysis and Parameter Design Optimization. John Wiley and Sons, INC., USA, 2000.

[Atkinson and Donev (1989)] A. C. Atkinson and A. N. Donev : The Construction of Exact D-optimum Experimental Designs with Application to Blocking Response Surface Designs. Biometrika, 76(3):515-526, 1989.

[Box et al. (1978)] G. E. P. Box, W. G. Hunter and J. S. Hunter : Statistics for Experimenters: An Introduction to Design, Data Analysis and Model Building. John Wiley and Sons, INC., New York, 1978.

[Cook and Nachtsheim (1980)] R. D. Cook and C. J. Nachtsheim : A Comparison of Algorithms for Constructing Exact D-Optimal Designs. Technometrics, 22(3):315-324, 1980.

[de Aguiar et al. (1995)] P. F. de Aguiar, B. Bourguignon, M. S. Khots, D. L. Massart and R. Phan-Than-Luu : D-optimal Designs. Chemometrics and Intelligent Laboratory Systems, 30(2):199-210, 1995.

[Donev and Atkinson (1988)] A. N. Donev and A. C. Atkinson : An Adjustment for the Construction of Exact D-Optimum Experimental Designs. Technometrics, 30(4):429-434, 1988.

[Dykstra (1971)] O. Dykstra : The Argumentation of Experimental Data to Maximize X'X. Technometrics, 13(3):682-688, 1971.

[Eriksson et al. (2000)] L. Eriksson, E. Johansson, N. Kettaneh-Wold, C. Wikström and S. Wold : Design of Experiments: Principles and Applications. Learnways AB, Umeå, 2000.

[Fedorov (1972)] V. V. Fedorov : Theory of Optimal Experiments (Review). Biometrika, 59(3):697-698, 1972. Translated and edited by W. J. Studden and E. M. Klimko.

[Galil and Kiefer (1980)] Z. Galil and J. Kiefer : Time- and Space-Saving Computer Methods, Related to Mitchell's DETMAX, for Finding D-Optimum Designs. Technometrics, 22(3):301-318, 1980.

[Johnson and Nachtsheim (1983)] M. E. Johnson and C. J. Nachtsheim : Some Guidelines for Constructing Exact D-Optimal Designs on Convex Design Spaces. Technometrics, 25(3):271-277, 1983.

[Meyer and Nachtsheim (1995)] R. K. Meyer and C. J. Nachtsheim : The Coordinate-Exchange Algorithm for Constructing Exact Optimal Experimental Designs. Technometrics, 37(1):60-69, 1995.

[Snee and Marquardt (1974)] Snee, R. D., and Marquardt, D. W.: A new algorithm for extreme vertices designs for linear mixture models. Technometrics, 16(2), 205-213.

[Scheffé (1958)(1963)] Scheffé(1958)(1963) .Experiments with Mixtures. Journal of the Royal Statistical Society: Series B (Methodological), 20(2), 344-360.

[Sinha (2014)] Sinha, D. (2014). Design and Analysis of Experiments with Mixtures. Wiley Series in Probability and Statistics. John Wiley Sons.

[Lehmann and Scheffé (1950)] Lehmann and Scheffé (1950): "Completeness, Similar Regions, and Unbiased Estimation. Part I." published in Sankhyā: The Indian Journal of Statistics (1933-1960), Vol. 10, No. 4 (1950), pp. 305-340.

[Lambrakis (1968)] Lambrakis, D. P. "Experiments with mixtures: A generalization of the simplex-lattice design." Journal of the Royal Statistical Society: Series B (Methodological) 30.1 (1968): 123-136.

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