



Course material

Course:

ENG606 / PHYS 442

Video:

DOE_lesson8_part1_FullFactorialDesign

Concepts (extracted from automatically generated subtitles):

Full factorial design. Matrix of experiments. Second degree coefficient. Main effects. Type of model. Analysis of experiments. Berman design. Type of matrices of structures. Special case of factorial design. Power n. First column. Second degree. Rule of the variance. Possible combinations of factor levels. Very good measurement.



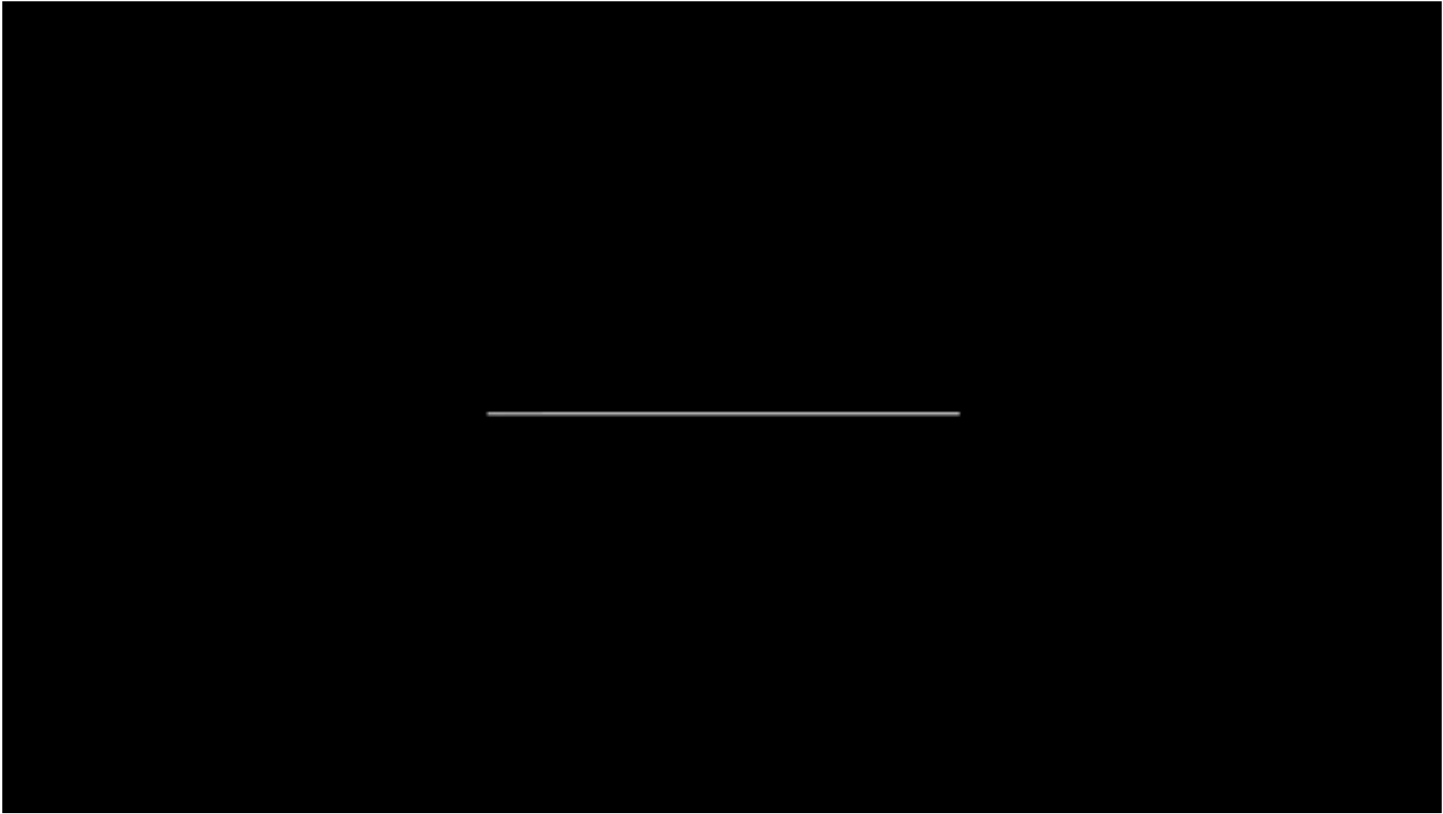
[to video sequence search](#)
(within ENG606 / PHYS 442.)



[to video](#)

Center for Digital Education. More educational support material here:

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>
page 1/33



...

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

.....

.....

0m 0s



.....

.....

what could say to students as an introduction to full factorial design considering that they already know about Plackett-Burman designs?

Memory updated

To introduce full factorial design to students who are already familiar with Plackett-Burman designs, you could emphasize how factorial designs offer a deeper exploration of factor interactions, building on their knowledge of screening designs like Plackett-Burman. Here's a suggested introduction:

"Now that you're familiar with Plackett-Burman designs, which are great for screening many factors with minimal experimental runs, let's move to full factorial design. Unlike Plackett-Burman, which focuses on identifying which factors are significant, a full factorial design goes further by exploring all possible combinations of factor levels. This allows us not only to identify main effects but also to see how factors interact with each other.

In a full factorial design, every factor is tested at each level across all other factor levels. This provides a comprehensive look at both main effects and interactions, including higher-order interactions if necessary. The trade-off here is efficiency: factorial designs require more experimental runs than Plackett-Burman. However, the depth of insight they offer is invaluable, especially when you're studying complex systems where interactions between variables could significantly impact the outcome.

Message ChatGPT



ChatGPT can make mistakes. Check important info.

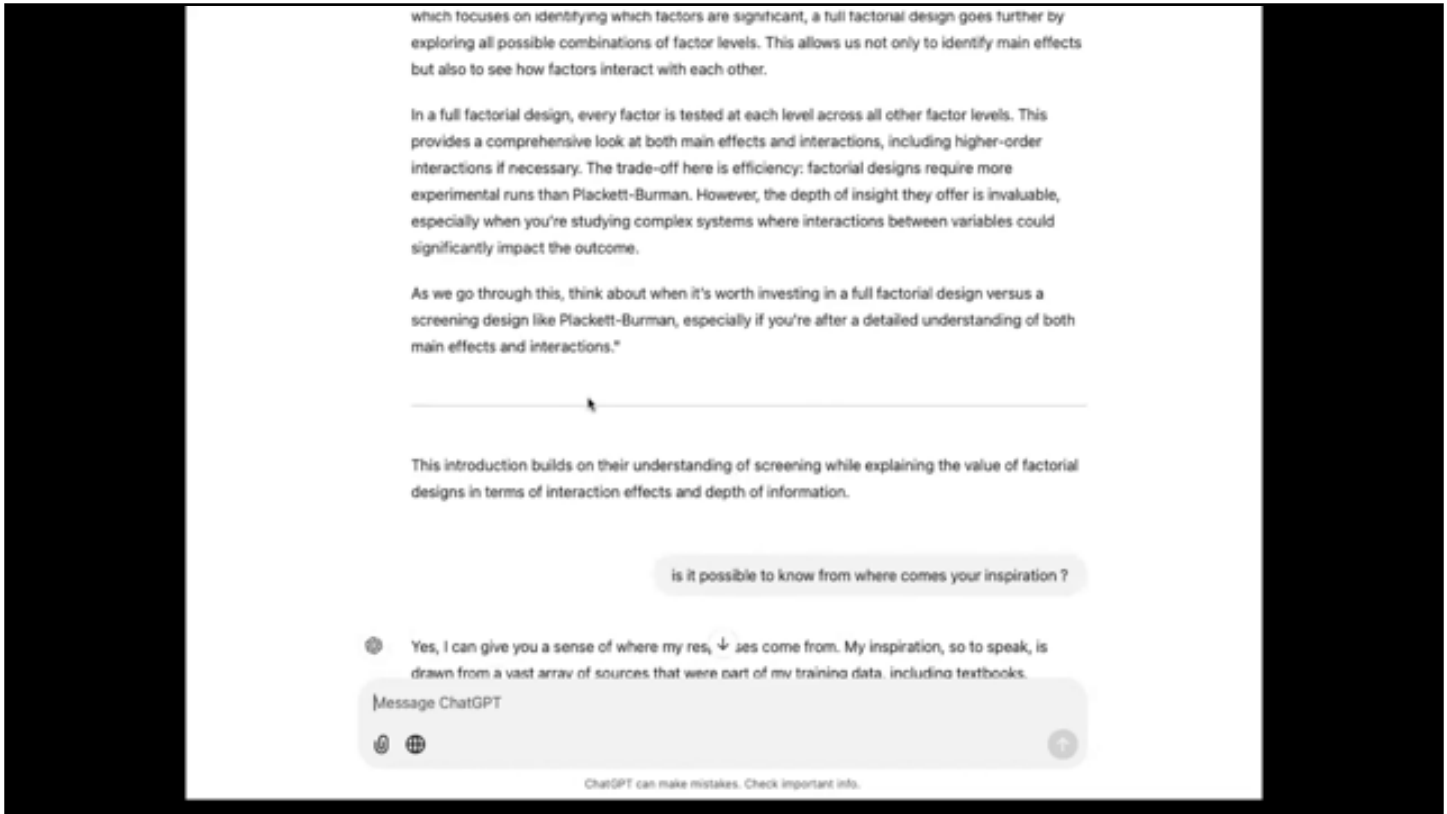
These subtitles have been generated automatically I was asking, chat GPT what to tell you as an introduction. And it's funny. So it's why I show you my dialogue with chat GPT.

notes

summary

0m 1s





And I already said with chat GPT, you better know the answer of your questions. But nevertheless, it's interesting. So I tell him, what could I say to my students about full factorial design if they already know about Placket and Berman design? He tells me a few things that in fact are quite OK. More or less what I have in mind. I never speak as nicely as chat GPT, but it's OK. Now that you are familiar with Placket and Berman design, which are great for screening many factors with minimal experimental runs, let's move to full factorial design. Men like Placket and Berman, which focuses on identifying which factors are significant, a full factorial design goes further by exploring all possible combinations of factor levels. This allows us not only to identify main effects, but also to see how factors interact with each other. In a full factorial design, every factor is tested at each level across all other factors levels. In fact, I will stay with two power N, so just two levels. But the words factorial designs, more largely when you test all the factor levels that you are interested. The trade-off here is efficiency. Factorial design require more experimental run than Placket and Berman. However, the depths of insights they offer is invaluable, especially when you are studying complex systems where interactions between variables could significantly impact the outcome.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



significantly impact the outcome.

As we go through this, think about when it's worth investing in a full factorial design versus a screening design like Plackett-Burman, especially if you're after a detailed understanding of both main effects and interactions."

This introduction builds on their understanding of screening while explaining the value of factorial designs in terms of interaction effects and depth of information.

is it possible to know from where comes your inspiration ?

Yes, I can give you a sense of where my responses come from. My inspiration, so to speak, is drawn from a vast array of sources that were part of my training data, including textbooks, academic papers, websites, and other resources on various topics. I also follow principles of structured, clear communication to craft responses tailored to your needs and background.

When I generate explanations or suggestions, I often rely on pedagogical frameworks and best practices in teaching complex topics, especially in areas like experimental design, statistics, and engineering. This approach is guided by a blend of subject knowledge, common teaching methods, and effective ways to break down ideas for different levels of expertise. I also consider context, such as your students' prior knowledge of Plackett-Burman designs, to make my answers relevant and actionable.

Message ChatGPT

ChatGPT can make mistakes. Check important info.

As we go through this, think about when it's worth investing a full factorial design versus a screening design like Placket and Berman, especially if you are after a detailed understanding on both main effects and interactions.

notes

summary

2m 3s



accessible.

any source specifically in this situation?

- For explaining full factorial designs to students who are already familiar with Plackett-Burman designs, there isn't a single, specific source that I directly reference, but rather a combination of general resources in experimental design and educational methods. Typically, I draw from core topics in experimental design and applied statistics that you would find in sources like:
1. **"Design and Analysis of Experiments"** by Douglas C. Montgomery: This is a foundational text widely used in teaching factorial designs and various experimental design types, including screening designs like Plackett-Burman and full factorial designs. Montgomery's explanations of when and why to use different designs would be relevant here.
 2. **"Statistics for Experimenters"** by George E.P. Box, J. Stuart Hunter, and William G. Hunter: This book emphasizes the logic behind choosing different experimental designs and explains how factorial and fractional factorial designs compare with screening designs in terms of identifying interactions.
 3. **Pedagogical principles for teaching experimental design:** Techniques in teaching advanced concepts, such as building on prior knowledge and using contrasting cases (in this case, contrasting screening and factorial designs), come from educational research and instructional design principles. Works in educational psychology, especially around "scaffolding" and "constructivism," help in shaping introductions that build on existing knowledge.

Message ChatGPT



ChatGPT can make mistakes. Check important info.

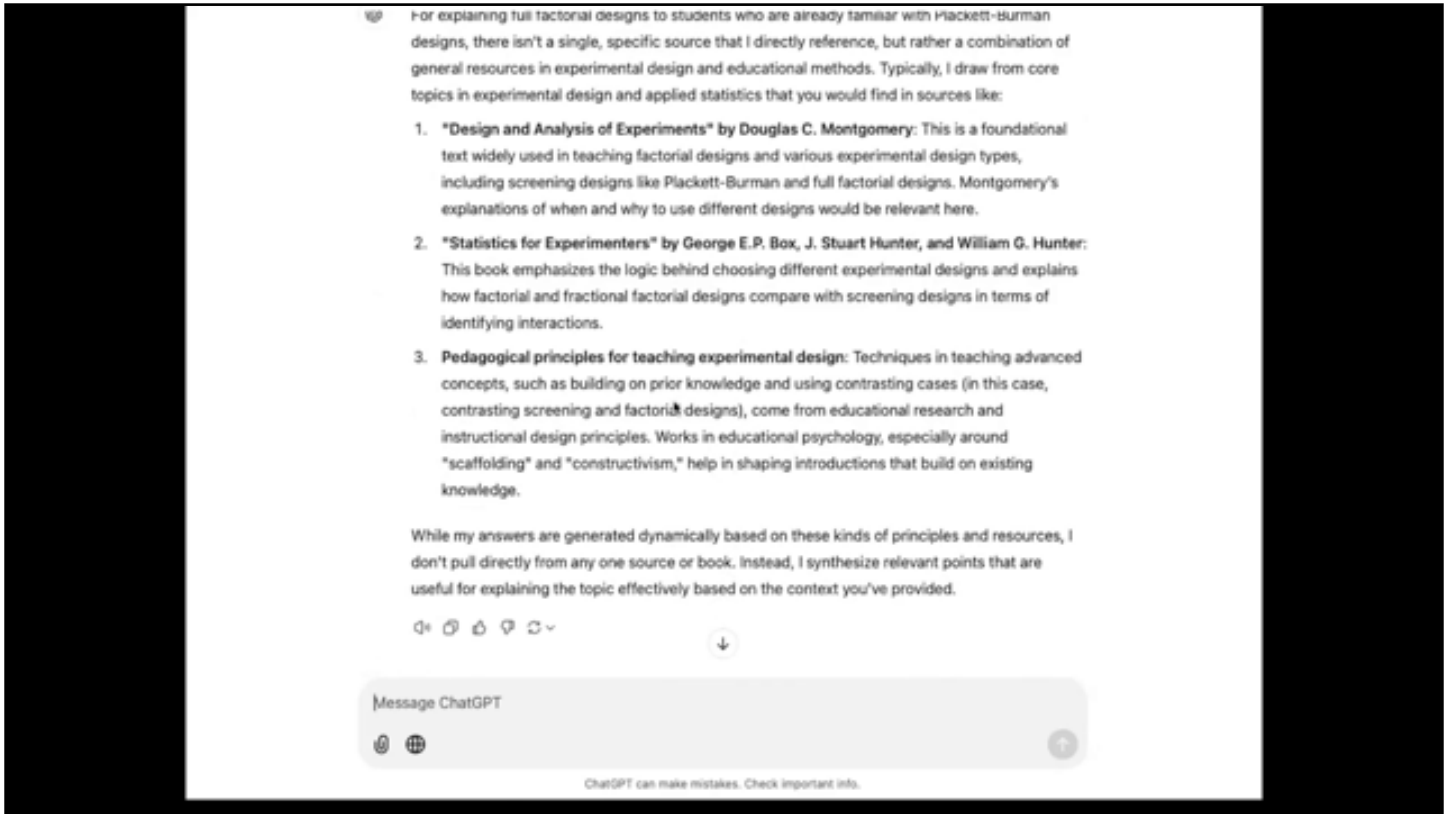
After he tells me that it's what I have to tell you, and after I have this question, is it possible to know where comes inspiration? So blah blah blah. Chagipiti gives me a few explanations how he creates answers, not so interesting.

notes

summary

2m 22s






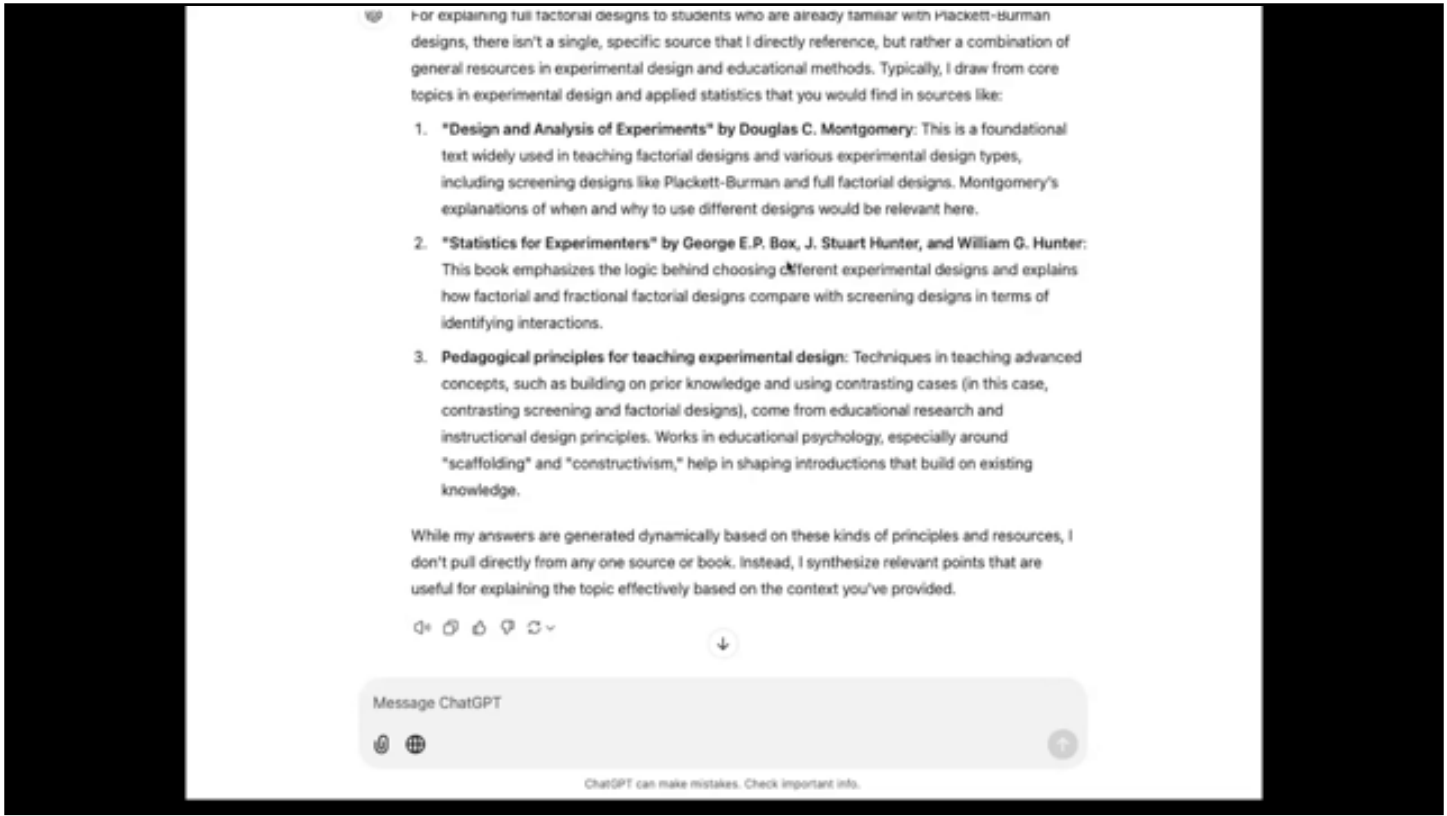
I ask him, any sources specifically in this situation?

notes

summary

2m 36s





He tells me, he quotes me two books and exactly my reference, so I don't know if he knows my course or if I'm so good at the reference I'm using, but he's proposing to use design and analysis of experiments from Douglas Montgomery and statistics for experiment for boxes.

notes

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

summary

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....



4.2 Full factorial design 2^n

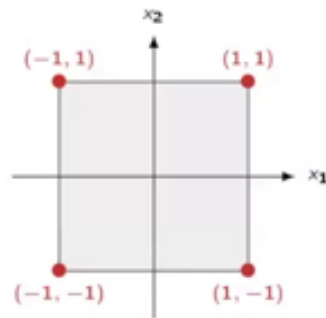
There are two bibles I'm using for this course. And he also talks about pedagogical principles. So I find it funny, so it's why I present it to you. So now you understand we are jumping to another chapter. So let's go back to my slides.

notes

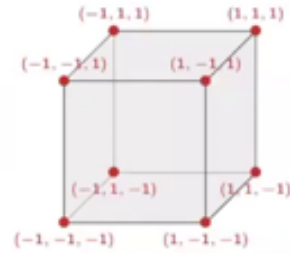
summary

3m 3s



4.2.1 Full factorial design 2^n 

$$E = \begin{pmatrix} -1 & 1 \\ -1 & -1 \\ 1 & 1 \\ 1 & -1 \end{pmatrix}$$



$$E = \begin{pmatrix} -1 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & 1 & -1 \\ -1 & 1 & 1 \\ 1 & -1 & -1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

Dr Jean-Marie Fürbringer

Modelling and design of experiments

So as I mentioned, factorial designs mean that you are using all the possible combinations of your factor. And I will present you today a special case of factorial design is the two power N factorial design. It exists also three power N that could be interesting for second degree. I will present them in the next chapter. But in this chapter, we keep to the two power N . Rapidly, you understand that two power N is the name of the design, but it's also the number of runs. So two power N makes four runs, eight runs, 16 runs, 32 runs, 64 runs, etc. So the increasing of the number of runs is exponential. This is the drawback of this design. Very powerful, but quite costly. So here you see the full factorial design for two dimension. You see that it's correspond to the corner of your domain if we have parallel epipedic domain. When the domain is not parallel epipedic, we have to perhaps diminish the knowledge so that we can draw within our domain a parallel epipedic subdomain. And when you look at the matrix of experiments, you see that you have a structure that will be all the time the same, half minus one, half plus one, a quarter minus one, a quarter plus one, a quarter minus one, a quarter.

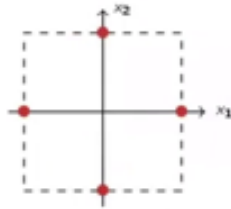
notes

summary

3m 22s

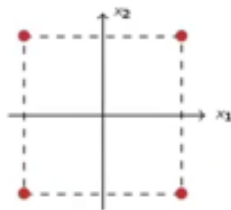


4.2.2 OFAT vs factorial design



$$\text{var}(\text{effet}) = 2\sigma^2$$

$$\text{si } \frac{\sigma}{\mu} = 10\% \rightarrow \frac{s}{m} = \frac{\sqrt{2}\sigma}{\mu} = 14\%$$



$$\text{var}(\text{effet}) = \frac{\sigma^2}{N}$$

$$\text{si } \frac{\sigma}{\mu} = 10\% \rightarrow \frac{s}{m} = \frac{\sigma}{\sqrt{4}\mu} = 5\%$$

This way of writing the matrix of experiments is called the canonical order. We start with what we can consider as the minimum point, which corresponds to the left lower corner of your domain. Then you finish with which is minus one, minus one, and we finish with the right highest corner plus one, plus one. It's not the objective of this course, but this type of matrices of structures also used for ordering elements when you have several dimensions. So it lets you go from the first one to the last one in three, four, five, six, seven dimensions very easily. So it's used for classification. When we jump to 3D, you see the same element. So the matrix of experiments, half minus one, half plus one, a quarter minus one, a quarter plus one, a quarter minus one, a quarter plus one, and a eighth minus one, a eighth plus one, etc., etc. And you continue in the other column. So you see also again, it corresponds to the corner of the domain. You could ask the question, is it sufficient to make measurements only at the border of the domain? Yes, as far as you are looking for linear responses. So now the type of model we will fit would be not only hyperplanes, but quite shifted hyperplane, that we do not have hills and valleys when we are looking in the direction of the axis. If you are changing your angle or point of view, it will correspond to some second degree surface because the model will have second degree coefficient, but only interactions, not pure quadratic terms. And even if it's analysis, it makes no difference in the research, in the empirical modeling, it makes a difference. It's not the same thing or considering interaction between factors or starting to consider quadratic influence of some factors. It's a step. It's a different step.

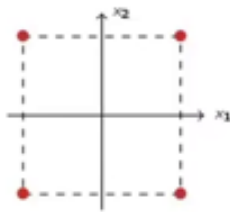
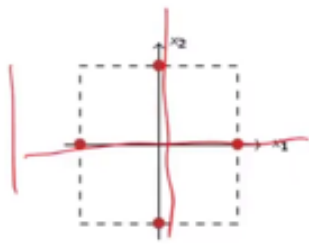
notes

summary

5m 25s



4.2.2 OFAT vs factorial design



Strategy

exp. var

$$\text{var}(\text{effet}) = 2\sigma^2$$

si $\frac{\sigma}{\mu} = 10\% \rightarrow \frac{s}{m} = \frac{\sqrt{2}\sigma}{\mu} = 14\%$

$$\text{var}(\text{effet}) = \frac{\sigma^2}{N}$$

si $\frac{\sigma}{\mu} = 10\% \rightarrow \frac{s}{m} = \frac{\sigma}{\sqrt{4}\mu} = 5\%$

Here is a small argument, easy just with the fingers, for showing you why it's interesting to use factorial design. I'm comparing here the OFAT, one factor at a time, design with a full factorial design for a very, very simple situation. So you see in the upper part of the slides is what I call a sort of OFAT. It's a star design. Again star design is my name. You will not find any literature on the star design. It's the fact that when you vary, you vary one factor after the other. So I vary first one factor and after the other factor. So if you do that, the effects of your factors due to the rule of the variance, the fact that when you have a difference, so you calculate the variance by making the difference between two measurements, like that you can estimate the slope of a factor. We do that with a difference and the variance of a difference is the sum of the variance of the terms is at the variance work. You cannot subtract variances. You can only add variances. So that means that finally the quality, the variance of your effects will be equivalent to two times sigma square. Here sigma square represents the experimental variance. Experimental variance. It will represent. So the sigma square represents your instrumentation, your technology, and the two is representing your strategy. This is a strategy. If you use factorial design, you can calculate more things and the different things you calculate are more precise. So you can start to calculate instructions, what you cannot do when you make one factor at a time.

notes

summary

8m 18s



4.2.3 What can we do with a 2^N factorial design ?

Identify the main effects and every interaction of n factors :

$$y = a_0 + \sum_i^n a_i X_i + \sum_{i < j}^n a_{ij} X_i X_j + \sum_{i < j < k}^n a_{ijk} X_i X_j X_k + \dots$$

The number of coefficients by level is

$$N_m = \binom{n}{m} = \frac{n!}{m!(n-m)!}, \quad m = 0 : n$$

And we know that $\sum_{m=0}^n \binom{n}{m} = 2^n = N_{exp}$

In MATLAB use the function `b=choosek(n,k)` to calculate the binomial coefficient

But especially now the variance, because you are using several couples for calculating the effect of the, in this case of the both variable is not two times sigma square is sigma square divided by four. There are numbers of experiments that you are doing. So in this example, so if you take the first one, if you consider you have a measurement with an accuracy of 10%, I mean the ratio between what you, the error of what you measure divided by the value of what you measure around 10%, which is not a very good measurement, but it's just for giving an idea. You finish finally for having an accuracy of your coefficients around 14%. If you do the same calculation with the same type of measurement with a factorial design, you finish with an accuracy of around 5%. So we have improved the accuracy quite three times. In fact, with the same technology, with better strategy, that was the same technology. And also, as I mentioned before, not only you can calculate with this strategy effects, but you can also calculate instructions, what would be not possible with the, of fact design.

notes

summary

11m 1s



4.2.4 2^N Factorial design on Matlab

Matlab

- ▶ Routine `fullfact()` generates a complete factorial plan depending on the number of indicated levels for each factor
- ▶ The levels begin at 1 → standardize the matrix
- ▶ `E=2*(fullfact([2 2 ...])-1.5)`
- ▶ Routine `ff2n()` generates a complete factorial plan with two levels for each factor
- ▶ The levels begin at 0 → standardize the matrix
- ▶ `E=2*(ff2n(nfact)-0.5)`
- ▶ `E=fracfact('a b c d')`

So what can we do with a factorial design, a two-end factorial design? So following what I already presented in a former chapter about the Taylor development and the Taylor polynomials, it's possible to evaluate the constant, the main effect, as it was the case in the Plackett and Berman design, but also instructions, the instructions two by two, but also the instructions three by three, five factors, four by four, five by five, six by six. So it could be quite interesting when you are in this situation with more than two factors to calculate the number of instruction coefficient. So the number of coefficient, the instruction coefficient is an arrangement. So it's the arrangement of n , the numbers of factor and m , the number of correlated coefficients. So you remember perfectly, I'm sure, you're the best student possible. So you remember that the arrangement is calculated by n factorial divided by m factorial and divided by the difference, the factor of the difference of n . And perhaps you remember, not kidding, I was remembering this rule, I have no idea why I'm remembering, but the sum of the arrangement is two power n . So you see it works very well. So with a factorial design, I don't know if it's magic or perhaps the magic of mass, you can evaluate all the interaction coefficients of the set of factors that you are, I don't know, using as factorial factors. If you want to make those type of calculations, remember the MATLAB or very probably the Python name of the function and Schuske is for calculating the arrangement when you want to make your own calculation. For this, it's called also the binomial coefficients. Okay, so a few indications and how you can build in MATLAB or you can build it by hand when you have two factors or three factors. It's not complicated to build the factorial design,

notes

summary

12m 37s



4.2.4 2^N Factorial design on Matlab

Matlab

- ▶ Routine `fullfact()` generates a complete factorial plan depending on the number of indicated levels for each factor
- ▶ The levels begin at 1 → standardize the matrix
- ▶ `E=2*(fullfact([2 2 ...])-1.5)`
- ▶ Routine `ff2n()` generates a complete factorial plan with two levels for each factor
- ▶ The levels begin at 0 → standardize the matrix
- ▶ `E=2*(ff2n(nfact)-0.5)`
- ▶ `E=fracfact('a b c d')`

the two n factorial design by hand or a small, very small algorithm. Nevertheless, it's useful sometimes to build. We will work with eight, seven, ten, eleven factors. So it's interesting for generating very rapidly those factors. So you have a routine in MATLAB, I don't know exactly in Python, but it's full fact. It's full factorial design because next sub-chapter will be fractional factorial design and so we will

notes

summary

4.2.5 Model matrix of a 2^n design

Matlab

- ▶ `X=x2fx(E,modelspec)`
- ▶ If `modelspec` is the keyword 'interactions' only the interactions of first order are computed
- ▶ To integrate more (or less) interaction coefficients $2 \times 2, 3 \times 3, \dots, n \times n$, the model must be defined by a coefficient matrix :

Handwritten notes: $a_0, a_1, a_{12}, a_{112}$

Handwritten note: $n \text{ coeff}$

Handwritten note: $\leftarrow \text{factors?}$

$$\text{modelspec} = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots \\ \text{etc} \end{pmatrix}$$

- ▶ Example :
`modelspec=[zeros(1,n); eye(N); unique(perms([1 1 0 ...], "rows")) : ...]`

learn to cut a little bit this design. So it's what we call them full facts, so two power and experiment. So full fact, but in this case, full fact is for several levels of factor. Could be two, three, four, five levels per factor. So you have to indicate to the routine that you want two levels. So it's what I do with this indication two, two, two. We are not limited to binary factors or to extremes. It's the same thing. We can continue to evaluate a mix in some situation, binary factors and or even with three levels or four levels and continuous factors. It's also an advantage of this type of design. We have to do it precisely, but we can do it. And this routine is giving values between if you have two levels, one and two, one, two, three, one, two, three, four. So you have to shift your results so that it will be, you have to normalize. So you have to shift your results to go between minus one and plus one. So if you have two level factor, you have to retract one point five because between one you have to go one and you have to multiply by two. It's exist another routine. Probably those routines have been done for not design of experiment, but for signal analysis. And it's why they are not fitting exactly to what we need. So FF2N, full factorial design 2N. This is also quite okay except that it's starting between zero and one. So we all again have to shift the matrix to go from minus one to plus one. So we have to shift it over one half and again to multiply by two for getting exactly the matrix that we want. I will also, it exists also one routine that do the job, but for explaining it, we need to understand the

notes

summary

16m 25s



4.2.5 Model matrix of a 2^n design

Matlab

- ▶ $X = x2fx(E, modelspec)$
- ▶ If *modelspec* is the keyword 'interactions' only the interactions of first order are computed
- ▶ To integrate more (or less) interaction coefficients $2 \times 2, 3 \times 3, \dots, n \times n$, the model must be defined by a coefficient matrix :

Handwritten notes: $a_0, a_1, a_{12}, a_{112}$

Handwritten note: $n \text{ coeff}$

Handwritten note: $\leftarrow \text{factors?}$

$$modelspec = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & 0 & \dots \\ \text{etc} \end{pmatrix}$$

- ▶ Example :
`modelspec = [zeros(1,n); eye(N); unique(perms([1 1 0 ...], "rows")) ; ...]`

next chapter because it's based on fractional factorial design. But nevertheless, it's a more straightforward way of generating in MATLAB the fractional, the full factorial design is using the `fact` that in the generator, I will explain later what is this generator network, but we have to give the number, the names of the different factors and no mixture between them. So it's why I have A, B, C, D. So the last comment will generate four power two full factorial design. When you have your model, your design matrix, your matrix of experiment, you would like to build the matrix of the model. So it takes this one function `x2fx`. So understand the name of the function. So from `x`, you make a function of `x`, `x` being a matrix. And so you give as argument the matrix of experiment `E` and you give the type of model that you want. So if you give nothing, it will make a linear model. So I probably, no, I didn't mention already in the previous subchapter, but so no model spec, make one column with 11111 for the constant and the linear factor. You can use keywords `linear`, the same thing, same thing as to think no, no, no comments. Interactions, but interactions give you the only the two by two interaction doesn't give you the highest order of interaction. And if you want all or thumb of the instructions, you better give to the routine a matrix. We call it the matrix of coefficient is something specific to MATLAB. It's a trick of programming. It's not an element of the statistic. And then how to understand. So this matrix of the model have one row per coefficient. So for each coefficient is a way to be described. So the constant, sorry, you have one row per coefficient and one column per factor. And you indicate the factor

notes

summary

4.2.5 Model matrix of a 2^n design

Matlab

- ▶ $X = x2fx(E, modelspec)$
- ▶ If *modelspec* is the keyword 'interactions' only the interactions of first order are computed
- ▶ To integrate more (or less) interaction coefficients $2 \times 2, 3 \times 3, \dots n \times n$, the model must be defined by a coefficient matrix :

Handwritten notes: $a_0, a_1, a_{12}, a_{112}$

Handwritten note: $modelspec =$

$$modelspec = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots \\ \text{etc} \end{pmatrix}$$

Handwritten note: \uparrow 17 coeff

Handwritten note: \leftarrow factors?

- ▶ Example :
`modelspec = [zeros(1,n) ; eye(N) ; unique(perms([1 1 0 ...], "rows")) ; ...]`

when they are considered in your coefficient. So the coefficient a zero has no coefficient, no factors involved is exactly what is the constant. So it will be zero, zero, zero, zero, all zero. That means no factor involved in this coefficient. When you want to describe to indicate that you want the factor a one, you put a one in the first column corresponding to the first factor and zero everywhere else. When you want the second one, the third one, you put one in the corresponding column. When you want to indicate that you are interested by an interaction coefficient, for example, a one two, you put a one in the first column, a one in the second column and zero the rest. So now you have understand. When you want the second degree of one coefficient, you put a two in the column for indicating the degree of this consideration. Now you can go like that with two, three, four, five, you can go up. So it's described a polynomial model and each row correspond to one term and you indicate what you have in the term if you would be interested by the coefficient a one, one, two, that will be a coefficient of the third degree with the quadratic influence of the first one and just a simple influence of second one is a sort of interaction coefficient. So you will write a two is the first column and the one in the second column and zero is the rest of the columns. And after the size of your matrix, if you have three factors, you have three column, if you have four factors, four columns, so you adapt the size of your. So you can write it by hand. It's what we usually do when the model is small. You can also find a way of computing, calculating this matrix is an example I

notes

summary

4.2.5 Model matrix of a 2^n design

Matlab

- ▶ `X=x2fx(E,modelspec)`
- ▶ If `modelspec` is the keyword 'interactions' only the interactions of first order are computed
- ▶ To integrate more (or less) interaction coefficients $2 \times 2, 3 \times 3, \dots n \times n$, the model must be defined by a coefficient matrix :

Handwritten notes: $a_0, a_1, a_{12}, a_{112}$

$$\text{modelspec} = \begin{pmatrix} 0 & \dots & \dots & \dots & 0 \\ 1 & 0 & \dots & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ 1 & 1 & 1 & 0 & \dots \\ \text{etc} \end{pmatrix}$$

Handwritten note: \uparrow 7 coeff

- ▶ Example :

`modelspec=[zeros(1,n); eye(N); unique(perms([1 1 0 ...], "rows")); ...]`

Handwritten note: \leftarrow factors?

play with it. So the first row was only zero, so it will zero one and it's a way of writing a line of zeros. And after if I want all the main effects, I have put the identity matrix. So I n, I e, y, e is the identity matrix, one in the diagonal. And after if you want. Instruction, if you want all the two by two instruction, so I start by defining a series. With an instruction of two, one, one, zero and the numbers of zero that you need. And after I calculate the permutations of all of this, but if you do that, you will get repetition in the permutation. So it's necessary to use this function unique that take out from a series of row. But you have to tell that you are selecting rows. You can also select columns. So it's an interesting function. So with this, you can build and two by two, three by three, four by four that you want to have in your model. So again, linear only the main effect and the constant quadratic. You add second degree coefficient. Interaction, you add only the two by two instructions or you can made it by yourself. Eventually, if you look in the help of this routine, you can perhaps write the Wilkinson notation for having it.

notes

summary

4.2.6 Case of a chemical reactor



Dr Jean-Marie Fürbringer

Modelling and design of experiments

I never done it, but probably you can also give the order to this routine with the Wilkinson models.

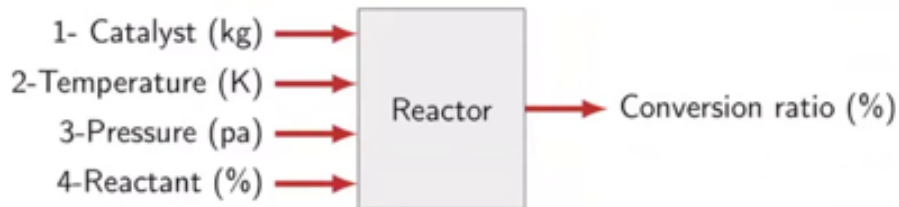
notes

summary

27m 25s



4.2.7 Block diagram



Okay, so let's now see an example for understanding how we use that. So sorry for the physicist. It's an example of chemistry, but please stay in my course. This will not be too complicated. So imagine that you have a reaction. So you have a reactor. So what is a reactor is volume, which is controlled in temperature and pressure. And you have input for injecting products, getting out product, etc. So the chemical reaction was okay, but we are now interested of increasing the size of the preparation. And we are making industrial tests for checking what could be better, controlling the temperature, controlling the pressure, controlling the quantity of different products for optimizing the production of a product.

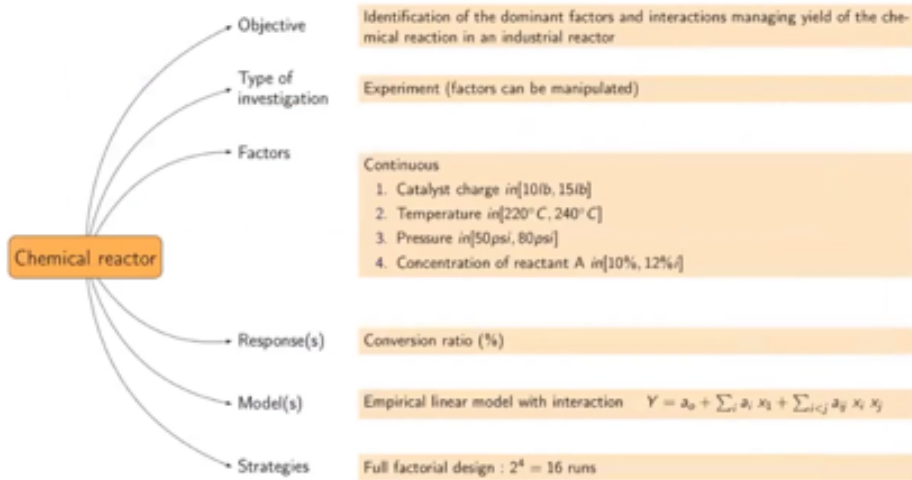
notes

summary

27m 28s



4.2.8 Mind map



So in what this comes from, the knowledge from the chemists or the different factors that could be interesting to optimize are the catalyst or products that we put in the reaction for helping the reactions to happen. The temperature, we can control the temperature of the reactor, the pressure, and the percentage of the reactant in the recipe. We will look at recipe when you have several products and you have to equilibrate the different products. In this case, we just have one. So it's easy. It's like another factor in this case is the percentage of the reaction. And as an answer, as a response, we have the conversion ratio. So if you are producing aspirin, it will be the quantity of aspirin that is produced after, let's say, an hour. Let's fix the time, an hour of the reaction. So this is the reaction that's happened in this reactor. We have three factors that we want to control and move. Eventually, we have other factors that we have controlled, but they are fixed values. And we have one answer that we would like to optimize, or at least we would like to understand what are the factors that manage this conversion ratio.

notes

summary

28m 32s



4.2.9 : Essay Matrix $2^4 \rightarrow 16 \times 4$

Standardized matrix			
x_1	x_2	x_3	x_4
-1	-1	-1	-1
-1	-1	-1	1
-1	-1	1	-1
-1	-1	1	1
-1	1	-1	-1
-1	1	-1	1
-1	1	1	-1
-1	1	1	1
1	-1	-1	-1
1	-1	-1	1
1	-1	1	-1
1	-1	1	1
1	1	-1	-1
1	1	-1	1
1	1	1	-1
1	1	1	1

Matrix with laboratory values			
Catalyst	Temperature	Pressure	Reactant A
[lb]	[° C]	[psi]	[-]
10	220	50	10%
10	220	50	12%
10	220	80	10%
10	220	80	12%
10	240	50	10%
10	240	50	12%
10	240	80	10%
10	240	80	12%
15	220	50	10%
15	220	50	12%
15	220	80	10%
15	220	80	12%
15	240	50	10%
15	240	50	12%
15	240	80	10%
15	240	80	12%

Matlab

`E=fracfact('a b c d')`

Dr Jean-Marie Fürbringer

Modelling and design of experiments

So let's draw a mind map. So the objective of this situation is to identify the dominant factors and interaction. What type of investigation is experiment because we are able to move those factors? We are not only observing what's happened in a series of tests. We are just changing the factors. So it's an experiment. We have four factors. In this case, there are four continues. We have the catalyst charge, which is between 10 pounds and 15 pounds. We have the temperature, which is between 220 and 240 degrees. And we have the pressure. I have put the imperial values for the pressure. You see, it doesn't change things. The same thing as when you are using international values. So between 50pc and 80pc, and we have the concentration of one of the reactants, which is between 10% and 12%. The response is the conversion ratio. And we would like to use an empirical model. Eventually it exists. Some reaction models, some calculation, but it's quite complicated models. So we prefer to use an empirical model for trying to optimize the production. So A0, the constant, main effect, AXX, YYX. AXXI and AIG and XIXJ. And the strategy that we will use is a full factorial design. 2 power 4. 2 because we are using two levels. 4 because we have four factors. So it represents 16, 16 runs. 2 power 4 equals 16. So we have the design, but it's also the number of runs that we are. So I select the routine of MATLAB or any other means. The matrix, the factorial matrix 2 power N. You see that it is organized as mentioned previously. Half minus one, half one. Quarter minus one, quarter one, quarter minus one, quarter.

notes

summary

30m 6s



4.2.10 Model Matrix (16×11)

$$y = a_0 + \sum a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \epsilon \quad (1)$$

i	x_1	x_2	x_3	x_4	$x_1 x_2$	$x_1 x_3$	$x_1 x_4$	$x_2 x_3$	$x_2 x_4$	$x_3 x_4$
1	-1	-1	-1	-1	1	1	1	1	1	1
1	-1	-1	-1	1	1	1	-1	1	-1	-1
1	-1	-1	1	-1	1	-1	1	-1	1	-1
1	-1	-1	1	1	1	-1	-1	-1	-1	1
1	-1	1	-1	-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1
1	-1	1	1	-1	-1	-1	1	1	-1	-1
1	-1	1	1	1	-1	-1	-1	1	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1	1
1	1	-1	-1	1	-1	-1	1	1	-1	-1
1	1	-1	1	-1	-1	1	-1	-1	1	-1
1	1	-1	1	1	-1	1	1	-1	-1	1
1	1	1	-1	-1	1	-1	-1	-1	-1	1
1	1	1	-1	1	1	-1	1	-1	1	-1
1	1	1	1	-1	1	1	-1	1	-1	-1
1	1	1	1	1	1	1	1	1	1	1

Matlab

```
X = x2fx(E,'interactions')
```

Eight and after the 16th changing all the time. But again, I'm insisting you never realize the design in that order. Because if you do that and you have something happening in your experiments, you have a product which is oxidizing and you have a regular change. You have the temperature increasing in your room, etc. You will have a NALIS between your problem and the effect. So we all the time execute these designs at random order. Another thing, you do not give this matrix, the standardized, the normalized matrix to your operator. It will not do what to do. So you have to translate it in the matrix used in a laboratory. So you have to give the value of the catalyst, of the temperature, of the pressure and the reactant. And you see the structure is the same except that no, it's not as normalized matrix. It is a real matrix with values between 10 pounds and 15 pounds, between 220 degrees centigrade and 240, between 50 and 80. See the pressure and the reactant between 10 and 12 percent.

notes

summary

33m 1s



4.2.10 Model Matrix (16×11)

$$y = a_0 + \sum a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \epsilon \quad (1)$$

1	x_1	x_2	x_3	x_4	$x_1 x_2$	$x_1 x_3$	$x_1 x_4$	$x_2 x_3$	$x_2 x_4$	$x_3 x_4$
1	-1	-1	-1	-1	1	1	1	1	1	1
1	-1	-1	-1	1	1	1	-1	1	-1	-1
1	-1	-1	1	-1	1	-1	1	-1	1	-1
1	-1	-1	1	1	1	-1	-1	-1	-1	1
1	-1	1	-1	-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1
1	-1	1	1	-1	-1	-1	1	1	-1	-1
1	-1	1	1	1	-1	-1	-1	1	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1	1
1	1	-1	-1	1	-1	-1	1	1	-1	-1
1	1	-1	1	-1	-1	1	-1	-1	1	-1
1	1	-1	1	1	-1	1	1	-1	-1	1
1	1	1	-1	-1	1	-1	-1	-1	-1	1
1	1	1	-1	1	1	-1	1	-1	1	-1
1	1	1	1	-1	1	1	-1	1	-1	-1
1	1	1	1	1	1	1	1	1	1	1

Matlab

```
X = x2fx(E,'interactions')
```

You can also build the matrix of the model. So in this case, the matrix of model will have one column that I'm calling I for identity, which correspond to the constant.

notes

summary

34m 36s



4.2.11 Measurement



I have four columns corresponding to my matrix of experiments, also representing my main effects. And I have in this case, a few column of interaction. In this case, we are, I've just written here the interactions two by two. I will come back on this aspect why we are using two by two and sometimes more interactions. If you want the full matrix, you have to add also the column for the instructions three by three and the one column of the instructions four by four for having the full. And you have 16 runs. And if you have all the coefficient, you will have also 16 columns. Here you have the way of building this routine.

notes

summary

34m 56s



4.2.10 Model Matrix (16×11)

$$y = a_0 + \sum a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \epsilon \quad (1)$$

1	x_1	x_2	x_3	x_4	$x_1 x_2$	$x_1 x_3$	$x_1 x_4$	$x_2 x_3$	$x_2 x_4$	$x_3 x_4$
1	-1	-1	-1	-1	1	1	1	1	1	1
1	-1	-1	-1	1	1	1	-1	1	-1	-1
1	-1	-1	1	-1	1	-1	1	-1	1	-1
1	-1	-1	1	1	1	-1	-1	-1	-1	1
1	-1	1	-1	-1	-1	1	1	-1	-1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1
1	-1	1	1	-1	-1	-1	1	1	-1	-1
1	-1	1	1	1	-1	-1	-1	1	1	1
1	1	-1	-1	-1	-1	-1	-1	1	1	1
1	1	-1	-1	1	-1	-1	1	1	-1	-1
1	1	-1	1	-1	-1	1	-1	-1	1	-1
1	1	-1	1	1	-1	1	1	-1	-1	1
1	1	1	-1	-1	1	-1	-1	-1	-1	1
1	1	1	-1	1	1	-1	1	-1	1	-1
1	1	1	1	-1	1	1	-1	1	-1	-1
1	1	1	1	1	1	1	1	1	1	1

Matlab

```
X = x2fx(E,'interactions')
```

After the work of the laboratory, you have the measurement. So we have finished.
We do not spend time analyzing the design.

notes

summary

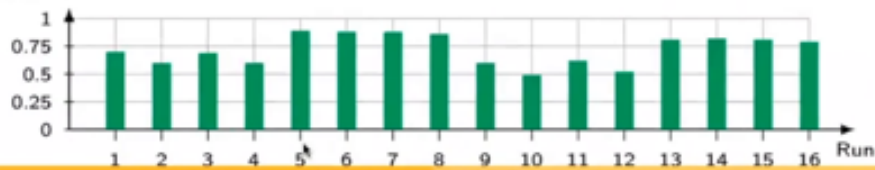
35m 55s



4.2.11 Case 8 : Results of the measurement

Run	Order	Catalyst	Temp.	Pressure	Reactant	Conversion [%]
1	9	-1	-1	-1	-1	70
2	16	-1	-1	-1	1	60
3	15	-1	-1	1	-1	69
4	3	-1	-1	1	1	60
5	10	-1	1	-1	-1	89
6	11	-1	1	-1	1	88
7	1	-1	1	1	-1	88
8	6	-1	1	1	1	86
9	2	1	-1	-1	-1	60
10	5	1	-1	-1	1	49
11	9	1	-1	1	-1	62
12	12	1	-1	1	1	52
13	4	1	1	-1	-1	81
14	14	1	1	-1	1	82
15	13	1	1	1	-1	81
16	7	1	1	1	1	79

Conversion ratio



Dr Jean-Marie Fürbringer

Modelling and design of experiments

We know that the variance inflation factor are one. We know that all the columns are orthogonal. So it's that aspect in the statistical aspect. This design is perfect. You cannot find better is like for Plackett and Bremen. If it's exactly what you want to want the main effect and the instruction is the best one. You do not have better situation, better design. So we can jump to the measurements.

notes

summary

36m 6s



4.2.12 Effect inference

$$\hat{\alpha} = \frac{1}{16} X'Y$$

Coefficient	Estimator	Estimate
α_0	1	72.25
α_1	1	-4.00
α_2	2	12.00
α_3	3	-0.13
α_4	4	-2.75
α_{12}	12	0.50
α_{13}	13	0.37
α_{14}	14	0.00
α_{23}	23	-0.62
α_{24}	24	2.20
α_{34}	34	-0.13
α_{123}	123	-0.38
α_{124}	124	0.25
α_{134}	134	-0.13
α_{234}	234	-0.38
α_{1234}	1234	-0.13

So you realize so you see here that you have the run and you have the order in which they have been realized. Be careful of attributing correctly the right response to the right experiment. And when you do that so you can make check with bar charts to see your results. You see the variation, but at that level you don't understand a lot. After you can calculate the coefficient of your model and the calculation can be done with simplified algorithm. Because in fact the full factorial matrices are matrix of Adamar. I'm just talking about the mathematical structure called matrix of Adamar. That means that the product of the matrix which is transposed make a diagonal matrix with the number of elements on the diagonal. When you reverse it will be one divided by n. So this is the simplified least square feet algorithm. So if you do that you can get the different coefficient. In this case you see I have all the possible coefficient, the constant, the main effect, the interaction 2 by 2, the interactions 3 by 3, the interaction 4 by 4.

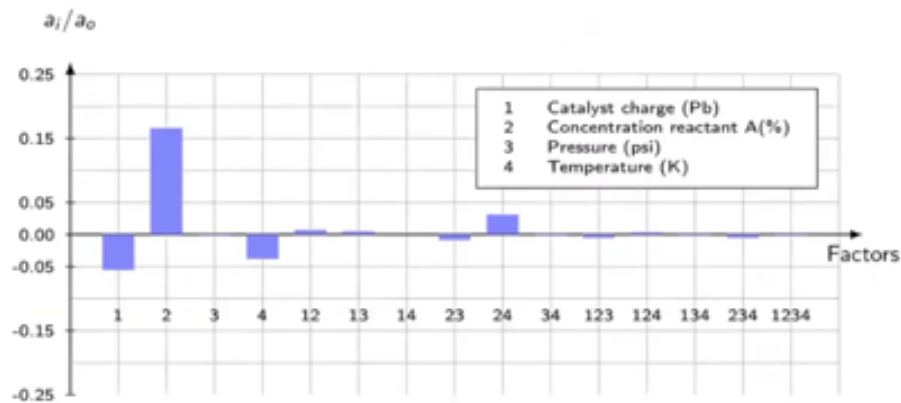
notes

summary

36m 37s



4.2.13 Relative half-effects



Matlab

```
alpha = X' * Y / Nexp
alpha_relative = alpha / alpha(1)
```

Dr Jean-Marie Fürbringer

Modelling and design of experiments

Forget the column estimator and if you go on the estimate you have the value of this. So you can get this number by this calculation, matrix calculation, what's 16 of x transpose y. You can also use fit L.M. algorithm. If you do that with the model with 16 coefficient, you have no degrees of freedom. So it will give you the values of the coefficient but the statistical in the ANOVA and the p values will be infinity or one half. You will have trouble on that part of the algorithm because you have no degrees of freedom if you want. 16 coefficient with 16 experiments, no degrees of freedom. The routine has some problems calculating the p value. Hopefully.

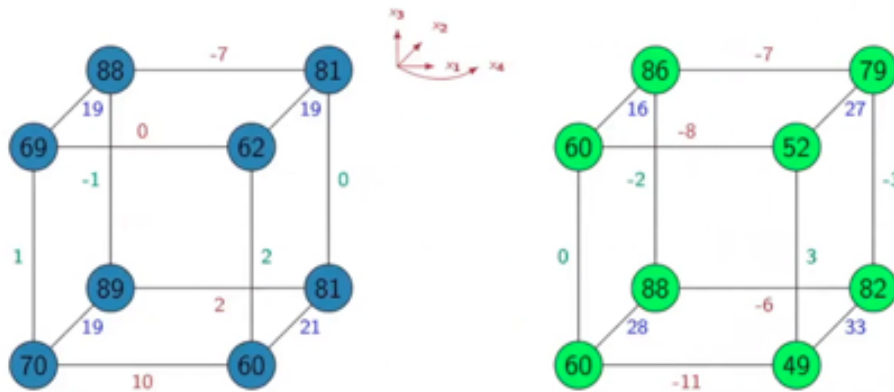
notes

summary

38m 25s



4.2.13 Visualization of the Effects



Dr Jean-Marie Fürbringer

Modelling and design of experiments

When I have that what I usually do, I make a bar chart and I usually do also a bar chart of what I call the relative coefficient. I treat the constant differently from the coefficient. You remember the constant represents the factor that you are not considering in your experiments. The factors, the coefficients represent the product of your job. You have various factors and you get the influence of those variations. And the residue represents what you are not managing but which is changing during your experiments. It's the noise, what we call the noise. So look. I have some coefficients. For one, I see that it's for the concentration reactants that I have the smaller one. So this seems to be important. This important, this important. The rest is just very, very small. For making a correct selection, you have to calculate variance and experimental variance that can let you select. But I will show you different ways of selecting what we call the significant factor. You will have value for all your coefficients. That's our way of selecting the ones that are significant. And the first approach, the biggest are significant, the smallest risk to be non-significant. The problem is the threshold when you can consider is significant or not. Mainly you would like effects that are at least three times bigger than the noise. That's the rule of thumb. So here you also see a typical situation and you have a few coefficients that are significant and very probably quite a lot that are non-significant. This is the problem, let's say, with this type of design. It's 16 experiments and in fact you see you have one, two, three, four coefficients that it's valid. You didn't know before, so you have learned and learning that something is not significant is an information. So could be you can argue that it's not a loss of energy

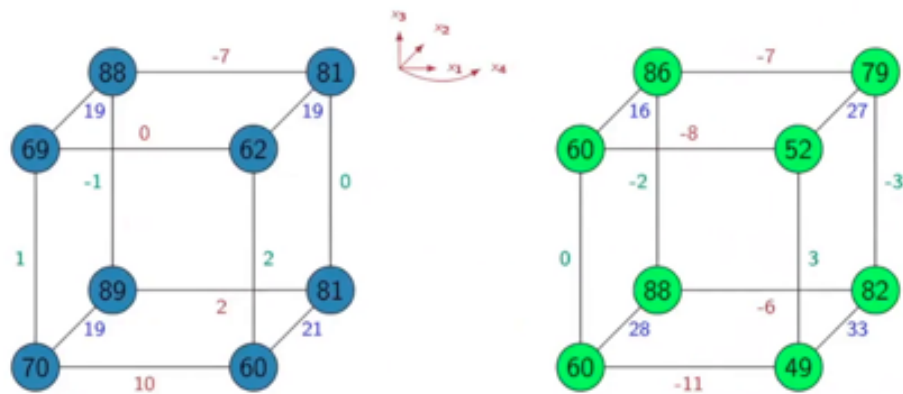
notes

summary

39m 13s



4.2.13 Visualization of the Effects



and time. But nevertheless, it's a pity that you have only, you have made 16 experiments for getting only four numbers. But it's also the price to pay for the quality.

notes

summary

