PLANOR : an R package for the automatic generation of regular fractional factorial designs

Version 1.0

A. Kobilinsky, A. Bouvier, H. Monod

October 4, 2012

INRA, UR 341, Unité MIAJ (Mathématiques et Informatique Appliquées - Jouy)
Institut National de la Recherche Agronomique
Domaine de Vilvert
F78352, Jouy en Josas Cedex
France

Contents

List of Figures

List of Tables

1 Introduction

The PLANOR R package provides functions to generate design fractions with possibly one or several block systems. It can be considered as an R version of the PLANOR software presented in [16], and this document is actually a direct adaptation of [16] for the R user. The method used in PLANOR is derived from the key matrix method [28], described in detail in [21], [22], [18], and more simply in [7]. This method produces designs called *regular* in which effects are either estimable independently or completely confounded.

In the simplest case, the user must provide the variance analysis model, specifying the terms sought to be estimated in this model. PLANOR searches for one or more designs meeting the users requirement among designs which can be constructed by this method. As there is not always a solution, the program indicates how far it has proceeded in the construction of the design, thus enabling the user to make a new request in order to obtain a solution.

PLANOR can take into account hierarchical constraints among factors. It is also possible to introduce several *"models"* together with the corresponding families of terms to be estimated, called *"parts to be estimated"*. This feature is particularly useful in a block design in which a factor must remain constant within each block. In such a case, its effect is not estimable within blocks, that is, in a model with a block effect. But it can be estimable between blocks if its estimation is requested in a model with no block effect: confusion is thus avoided with any other treatment effect. When there are several block systems, this feature can be used to require the estimation of certain effects in pre-specified *strata*.

PLANOR was initially designed to draw up experimental designs to operate a robot. This robot, developed by the Laboratoire du Génie de l'Hygiène et des Procédés Alimentaires de l'INRA Massy (LGHPA) (Laboratory for public health and food process engineering of INRA Massy) in the framework of a contract with the Association pour le développement de la Recherche dans l'Industrie LAITière (ARILAIT) (Association for the development of dairy industry research) is used to test surface cleaning and disinfection procedures. For this reason, a substantial part of this instruction manual is devoted to the designs for this robot, which in fact illustrate the program's abilities very well.

The analysis and interpretation of the designs obtained by PLANOR is usually simple, provided the user is quite familiar with the variance analysis techniques. In particular, when there are blocks and certain factors remain constant within the blocks, the analysis draws on notions of *inter- and intra-block strata*. The main effect of a factor remaining constant within each block is tested against *inter-block* variance, which differs from *intra-block* error variance used to test the other effects. Another case in which the analysis requires caution is when the number of *degrees of freedom* of the error is very low or zero. Additional indications and references on these subjects may be found in [20] and [18].

The presentation adopted in this instruction manual avoids algebraic formalism,

thus rendering the text a little less rigorous but thereby making it accessible to a much wider public. The slightly difficult passages are printed in small type and are preceded by an asterisk. They may be skipped without affecting understanding of the rest of the text.

2 Presentation of the method

Experimental units are identified by the levels of a certain number of factors, referred to as *basic factors*. From these factors, potentially decomposed into *pseudofactors*, the program defines new factors, also called *added*, *derived* or *defined* factors, satisfying the conditions imposed, if possible. These factors are linear combinations of the basic factors or pseudofactors resulting from their decomposition. The following few examples effectively illustrate the method and its properties.

2.1 Example with 2-level factors

2.1.1 Definition and properties of the design

There are 8 units identified by combinations of levels of 3 treatment factors A, B, C, each with 2 coded levels 0 and 1. From these three basic factors, a new factor D is defined by setting $D = A + B + C \pmod 2$. This same design can also be defined by the relation $D = ABC$ if the levels are coded as 1 and -1. In order to distinguish between the two codings and associated layouts, we refer to *additive* notation in the first case and to *multiplicative* notation in the second case. The shift from additive to multiplicative notation occurs by replacing each α level by level $(-1)^\alpha$. Table 1 presents the design in both its forms.

$D = A+B+C$				$D = ABC$				
A	B	C	D	A	B	C	D	ind.rep.
0	0	0	0	1	1	1	1	0
0	0	1	1	1	1	−1	−1	2
0	1	0	1	1	−1	1	−1	7
0	1	1	0	1	−1	−1	1	1
1	0	0	1	−1	1	1	−1	6
1	0	1	0	−1	1	−1	1	5
1	1	0	0	−1	−1	1	1	4
1	1	1	1	−1	−1	−1	−1	3

Table 1: Example 2.1: 4 factors and 2^3 experimental units.

The factorial effects studied are the main effects A, B, C, D of the factors and their interactions AB, AC, ..., $ABCD$. These effects are also noted additively: $e(A)$ for the main effect of A, $e(A+B)$, $e(A+C)$, ..., $e(A+B+C+D)$ for the interactions. The functional notation $e()$ with this additive notation is essential to distinguish a sum of effects such as $e(A+B)+e(C)$ from the corresponding interaction $e(A+B+C)$. It will sometimes also be used in multiplicative notation to distinguish an effect from the corresponding product of factors (equal to -1 or 1).

The precise definition of effects is simple in multiplicative notation. If we denote by $\tau(A, B, C, D)$ the mean response –in more statistical terms, the expectation of the

added term	deduced equality	aliased effects
	$0 = A + B + C + D$	$e(0), e(A + B + C + D)$
A	$A = B + C + D$	$e(A), e(B + C + D)$
B	$B = A + C + D$	$e(B), e(A + C + D)$
C	$C = A + B + D$	$e(C), e(A + B + D)$
$A + B$	$A + B = C + D$	$e(A + B), e(C + D)$
$A + C$	$A + C = B + D$	$e(A + C), e(B + D)$
$B + C$	$B + C = A + D$	$e(B + C), e(A + D)$
$A + B + C$	$A + B + C = D$	$e(A + B + C), e(D)$

Table 2: Aliased effects in the example 2.1.

response– for the (A, B, C, D) treatment, the main effect of A and the AB interaction are for instance defined by

$$\begin{aligned} e(A) &= \frac{1}{16} \sum_{A,B,C,D} A\, \tau(A, B, C, D) = \frac{1}{2} \sum_{A} A\, \tau(A, \bullet, \bullet, \bullet) \\ &= \frac{1}{2}\Big(\tau(1, \bullet, \bullet, \bullet) - \tau(-1, \bullet, \bullet, \bullet)\Big) \\ e(AB) &= \frac{1}{16} \sum_{A,B,C,D} AB\, \tau(A, B, C, D) = \frac{1}{4} \sum_{A,B} AB\, \tau(A, B, \bullet, \bullet) \\ &= \frac{1}{4}\Big(\tau(1, 1, \bullet, \bullet) - \tau(1, -1, \bullet, \bullet) - \tau(-1, 1, \bullet, \bullet) + \tau(-1, -1, \bullet, \bullet)\Big)\ . \end{aligned}$$

Each $(\bullet)$ point indicates that the mean has been determined from the corresponding letter. For example:

$$\tau(A, B, \bullet, \bullet) = \frac{1}{4}\Big(\tau(A, B, 1, 1) + \tau(A, B, 1, -1) + \tau(A, B, -1, 1) + \tau(A, B, -1, -1)\Big)$$

The defining relation $D = A + B + C (\text{mod } 2)$ is rewritten in the form

$$A + B + C + D = 0 \pmod 2\ . \tag{1}$$

By adding to this equality the sums $A, \ldots, A+B+C$ formed from the three basic factors, we obtain the equalities appearing in the second column of table 2. Through the following simple rule, the sets of *aliased* effects appearing in the third column of the table can then be obtained. Note that the terms *confounded* and *aliased* are synonymous in this context, therefore we shall use both terms indifferently.

Rule 1 *The effects corresponding to two equal sums α and β are aliased.*

Note that $e(0)$ is by definition the general mean. The interaction $e(A + B + C + D)$ is thus confounded with this general mean.

In this instance, the confounding of two effects, for example $e(A+B)$ and $e(C+D)$ results in the fact that only their sum $e(A+B)+e(C+D)$ can be estimated. The aliased effects are the same when 1 is added to the definition of D, that is, if $D = A+B+C+1$ (mod 2), or even

$$A+B+C+D = 1 \pmod 2, \quad (ABCD = -1 \quad \text{in multiplicative notation}) . \qquad (2)$$

In this case, it is not the sums but the differences, such as $e(A+B) - e(C+D)$, which are estimable.

Examination of the third column of table 2 shows that the main effects are counfounded in this example with 3-factor interactions and that two-factor interactions are confounded with each other. This results from the fact that the defining relation (1) contains 4 factors. Such a design is said to be of *resolution* 4. When 3-factor or 4-factor interactions are neglected, each of the sums $e(A)+e(BCD)$, ..., $e(ABC)+e(D)$ containing a main effect reduces to this main effect. The main effects are then estimable.

2.1.2 Principle of a design search by PLANOR

In the above, the properties of the design are inferred from its definition. PLANOR generally proceeds in reverse order. It searches for the design on the basis of the model and part to be estimated. Thus in this example, the first step is to specify the factors A, B, C, D and their level numbers. This is done by using the `planor.factors` function. Here, the result is stored in the R object `ABCD.fact`

```
> library("planor")
> ABCD.fact <- planor.factors(factors=c("A","B","C","D"),
+                             nlevels=rep(2,4))
```

The second step is to specify the model and part to be estimated, using the `planor.model` function and saving the result in another object `ABCD.mod`

```
> ABCD.mod <- planor.model(
+       model= ~ A + B + C + D + A:B + A:C + A:D + B:C + B:D + C:D,
+       estimate=~ A + B + C + D)
```

The third step is to search for a design of a given size that satisfies the model and estimated part that have just been specified. This involves searching for a special matrix called the design key and then using the design key to construct the actual design. The search for the design key is made by the function `planor.designkey` :

```
> ABCD.key <- planor.designkey(factors=ABCD.fact,
+                              model=ABCD.mod,
+                              nunits=8,
+                              base=~A+B+C)
```

```
Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for column 4 .
      first visit to column 4
The search is closed: max.sol =  1 solution(s) found
```

The results are stored in `ABCD.key` and the `base` argument is used to specify that A, B and C are the basic factors. The `planor.designkey` function explores the possibilities for D. It eliminates choices which do not enable estimation of the effects of A, B, C, D in the framework of the model considered. For instance, choice $D = AB$ is eliminated because it leads to confounding of the main effects A, B, D with interactions BD, AD, AB, respectively. In this instance, the only valid choice is $D = \pm ABC$, i.e. in additive notation, $D = A + B + C$ or $D = A + B + C + 1$.

The function `planor.design` uses the design key to construct the design. After selection at random or by the user of the constant 0 or 1 added to $A + B + C$ in order to obtain D[1], the design is constructed by default in the systematic order which appears in table 1. It is stored in the `design` slot of the output called here `ABCD.sys`.

```
> ABCD.sys <- planor.design(ABCD.key)
> print(ABCD.sys@design)
```

```
  A B C D
1 1 1 1 1
2 1 1 2 2
3 1 2 1 2
4 1 2 2 1
5 2 1 1 2
6 2 1 2 1
7 2 2 1 1
8 2 2 2 2
```

Note that the function `planor.design` replaces the level numbers 0 and 1 by the actual factor levels (1 and 2 by default), in order to obtain a design which may be readily used.

2.1.3 Randomization

The allocation of treatments to experimental units (plots in agriculture, animals in animal science, procedure number in a laboratory experiment, ...) is generally random. In the present case, there are no blocks and such random allocation, called *randomization*, is achieved by randomly drawing the unit number allocated to each of the 8 treatments. A

[1] not implemented

possible result of this draw is denoted by *ind.rep (repetition index)* in table 1. Getting a completely randomized rather than a systematic design is possible by using the argument `randomize=~UNITS` of the function `planor.design`, as shown in the following variant of the four-factor example :

```
> ABCD.factL <- planor.factors(factors=list(
+                                A=c("small","big"),
+                                B=c("cold","hot"),
+                                C=c("black","white"),
+                                D=c("yes","no")))
> ABCD.key <- planor.designkey(factors=ABCD.factL,
+                              model=ABCD.mod,
+                              nunits=8)

> ABCD.rand <- planor.design(ABCD.key, randomize=~UNITS)
> print(ABCD.rand@design)

      A    B     C   D
1   big cold white yes
2 small  hot black  no
3 small  hot white yes
4 small cold white  no
5 small cold black yes
6   big cold black  no
7   big  hot white  no
8   big  hot black yes
```

The same factors have been declared by `planor.factors`, except that explicit labels have been given to their levels. The output of `planor.design` is now a randomized design.

2.1.4 Alternatives for the writing of the model

`PLANOR` uses the R syntax for model formulae, which automatically completes the model by adding the constant and terms "included" in one of the interactions if the factors are separated by a star `*`. The writing of the above model can thus be shortened by omitting the main effects since each of the latter is included in an interaction:

```
model=~ A*B + A*C + A*D + B*C + B*D + C*D
```

When there are many factors, writing the model can become tedious. In order to shorten the process, brackets can be used. The model is developed and the redundant factors in each *effect* are deleted as well as the redundant terms in the formula.

Thus

$$\begin{array}{rcl}(A+B+C+D)(A+B+C+D) & \longrightarrow & A*A+A*B+A*C+A*D+B*A+B*B+B*C+B*D \\ & & +C*A+C*B+C*C+C*D+D*A+D*B+D*C+D*D \\ & \longrightarrow & A+B+A:B+C+A:C+D+A:D+B:C+B:D+C:D \\ & \longrightarrow & A+B+C+D+A:B+A:C+A:D+B:C+B:D+C:D\end{array}$$

therefore the previous model can be written in the form

`model=~(A+B+C+D)*(A+B+C+D)` or `model=~(A+B+C+D)^2`

If the model is intended to contain all except for one or two interactions, say $B.C$, $B.D$, it may be convenient to use an expression such as

`model=~(A+B+C+D)^2 - (B:C+B:D)` or[2] `model=~(A+B+C+D)^2 - B:(C+D)`

where the minus sign is used to subtract both interactions from the model.

2.2 Example with 3-level factors

2.2.1 Definition and properties

There are 27 units defined by 3 treatment factors A, B, C with 3 coded levels 0, 1, 2. A new treatment factor D and a block factor Bl are defined by setting $D = A + B + C$ and $Bl = A + B + 2C \pmod 3$.

The resulting design is given in table 3. The multiplicative notation appearing in the middle of this table uses the cube root j = $\exp(2\pi/3)$ of the unit, which replaces the -1 used in the case of 2-level factors.

The factorial effects of the main effect of A are A, A^2, and those of the interaction between A and B are AB, A^2B^2, AB^2, A^2B. The factorial effects of the other main effects and interactions are enumerated similarly. In additive notation, these same effects are noted $e(A)$, $e(2A)$, $e(A+B)$, $e(2A+2B)$, $e(A+2B)$, $e(2A+B)$, etc ... The precise definition of these effects, associated with a decomposition into orthogonal contrasts also referred to as orthogonal degrees of freedom, is by no means essential to understanding and using `PLANOR` .

By adding $2D$ to the equality $D = A + B + C$ and by then multiplying the result by 2 (mod 3), the following is obtained:

$$0 = A + B + C + 2D = 2A + 2B + 2C + D \; . \tag{3}$$

The addition of 27 linear combinations between the basic factors A, B, C then gives rise to the equalities appearing on the left side of table 4. The block effects Bl and $2Bl$ equal to $A + B + 2C$ and $2A + 2B + C$, respectively, have been added in brackets (caution: $2 \times 2\, C = 4C = C \bmod 3$).

[2] needs to be checked

$D = A + B + C$ $Bl = A + B + 2C$					$D = ABC$ $Bl = ABC^2$					Randomization (see table 5)	
A	B	C	D	Bl	A	B	C	D	Bl	Bl_0	ind-rep
0	0	0	0	0	1	1	1	1	1	2	2
0	0	1	1	2	1	1	j	j	j^2	0	6
0	0	2	2	1	1	1	j^2	j^2	j	1	6
0	1	0	1	1	1	j	1	j	j	1	0
0	1	1	2	0	1	j	j	j^2	1	2	1
0	1	2	0	2	1	j	j^2	1	j^2	0	1
0	2	0	2	2	1	j^2	1	j^2	j^2	0	7
0	2	1	0	1	1	j^2	j	1	j	1	2
0	2	2	1	0	1	j^2	j^2	j	1	2	5
1	0	0	1	1	j	1	1	j	j	1	8
1	0	1	2	0	j	1	j	j^2	1	2	7
1	0	2	0	2	j	1	j^2	1	j^2	0	8
1	1	0	2	2	j	j	1	j^2	j^2	0	2
1	1	1	0	1	j	j	j	1	j	1	7
1	1	2	1	0	j	j	j^2	j	1	2	0
1	2	0	0	0	j	j^2	1	1	1	2	4
1	2	1	1	2	j	j^2	j	j	j^2	0	4
1	2	2	2	1	j	j^2	j^2	j^2	j	1	4
2	0	0	2	2	j^2	1	1	j^2	j^2	0	3
2	0	1	0	1	j^2	1	j	1	j	1	5
2	0	2	1	0	j^2	1	j^2	j	1	2	8
2	1	0	0	0	j^2	j	1	1	1	2	6
2	1	1	1	2	j^2	j	j	j	j^2	0	0
2	1	2	2	1	j^2	j	j^2	j^2	j	1	3
2	2	0	1	1	j^2	j^2	1	j	j	1	1
2	2	1	2	0	j^2	j^2	j	j^2	1	2	3
2	2	2	0	2	j^2	j^2	j^2	1	j^2	0	5

Table 3: Example 2.2: 4 3-level factors on 3 blocks of 9 units

The rule 1 makes it possible to immediately infer from these equalities the sets of aliased effects, which appear on the right side of table 4.

The following properties of the design (which may be checked in table 4) are directly inferred from the equality in equation (3) :

- A main effect is confounded only with interactions of three or more factors.
- Among the four degrees of freedom of an interaction between two factors, two are aliased with another two-factor interaction, and two are only aliased with interactions of three or more factors.
- There is no set of three aliased effects which only contains three- or four-factor interaction effects.

The latter property shows that any other definition of Bl would make the block effect confounded with at least a two-factor interaction. However, the option selected here makes the interaction effects $e(C + D)$ and $e(2C + 2D)$ inestimable even though interactions of three or four factors are assumed to be null. A wiser choice would be $Bl = C + 2D$. This choice would enable confounding of the block effect with effects $e(C + 2D)$ and $e(2C + D)$ which are already confounded with other 2-factor interactions and are in any case inestimable.

The design thus provides the possibility of estimating all the main effects in the model including two-factor interactions and the block effect. The specifications resulting

equalities induced by (3)					aliased effects
	0	$=A+B+C+2D$	$=2A+2B+2C+D$		$e(0)$, $e(A+B+C+2D)$, $e(2A+2B+2C+D)$
	C	$=A+B+2C+2D$	$=2A+2B+D$		$e(C)$, $e(A+B+2C+2D)$, $e(2A+2B+D)$
	$2C$	$=A+B+2D$	$=2A+2B+C+D$		$e(2C)$, $e(A+B+2D)$, $e(2A+2B+C+D)$
	B	$=A+2B+C+2D$	$=2A+2C+D$		$e(B)$, $e(A+2B+C+2D)$, $e(2A+2C+D)$
	$B+C$	$=A+2B+2C+2D$	$=2A+D$		$e(B+C)$, $e(A+2B+2C+2D)$, $e(2A+D)$
	$B+2C$	$=A+2B+2D$	$=2A+C+D$		$e(B+2C)$, $e(A+2B+2D)$, $e(2A+C+D)$
	$2B$	$=A+C+2D$	$=2A+B+2C+D$		$e(2B)$, $e(A+C+2D)$, $e(2A+B+2C+D)$
	$2B+C$	$=A+2C+2D$	$=2A+B+D$		$e(2B+C)$, $e(A+2C+2D)$, $e(2A+B+D)$
	$2B+2C$	$=A+2D$	$=2A+B+C+D$		$e(2B+2C)$, $e(A+2D)$, $e(2A+B+C+D)$
	A	$=2A+B+C+2D$	$=2B+2C+D$		$e(A)$, $e(2A+B+C+2D)$, $e(2B+2C+D)$
	$A+C$	$=2A+B+2C+2D$	$=2B+D$		$e(A+C)$, $e(2A+B+2C+2D)$, $e(2B+D)$
	$A+2C$	$=2A+B+2D$	$=2B+C+D$		$e(A+2C)$, $e(2A+B+2D)$, $e(2B+C+D)$
	$A+B$	$=2A+2B+C+2D$	$=2C+D$		$e(A+B)$, $e(2A+2B+C+2D)$, $e(2C+D)$
	$A+B+C$	$=2A+2B+2C+2D$	$=D$		$e(A+B+C)$, $e(2A+2B+2C+2D)$, $e(D)$
$[Bl]=$	$A+B+2C$	$=2A+2B+2D$	$=C+D$		$[e(Bl)], e(A+B+2C), e(2A+2B+2D), e(C+D)$
	$A+2B$	$=2A+C+2D$	$=B+2C+D$		$e(A+2B)$, $e(2A+C+2D)$, $e(B+2C+D)$
	$A+2B+C$	$=2A+2C+2D$	$=B+D$		$e(A+2B+C)$, $e(2A+2C+2D)$, $e(B+D)$
	$A+2B+2C$	$=2A+2D$	$=B+C+D$		$e(A+2B+2C)$, $e(2A+2D)$, $e(B+C+D)$
	$2A$	$=B+C+2D$	$=A+2B+2C+D$		$e(2A)$, $e(B+C+2D)$, $e(A+2B+2C+D)$
	$2A+C$	$=B+2C+2D$	$=A+2B+D$		$e(2A+C)$, $e(B+2C+2D)$, $e(A+2B+D)$
	$2A+2C$	$=B+2D$	$=A+2B+C+D$		$e(2A+2C)$, $e(B+2D)$, $e(A+2B+C+D)$
	$2A+B$	$=2B+C+2D$	$=A+2C+D$		$e(2A+B)$, $e(2B+C+2D)$, $e(A+2C+D)$
	$2A+B+C$	$=2B+2C+2D$	$=A+D$		$e(2A+B+C)$, $e(2B+2C+2D)$, $e(A+D)$
	$2A+B+2C$	$=2B+2D$	$=A+C+D$		$e(2A+B+2C)$, $e(2B+2D)$, $e(A+C+D)$
	$2A+2B$	$=C+2D$	$=A+B+2C+D$		$e(2A+2B)$, $e(C+2D)$, $e(A+B+2C+D)$
$[2Bl]=$	$2A+2B+C$	$=2C+2D$	$=A+B+D$		$[e(2Bl)], e(2A+2B+C), e(2C+2D), e(A+B+D)$
	$2A+2B+2C$	$=2D$	$=A+B+C+D$		$e(2A+2B+2C)$, $e(2D)$, $e(A+B+C+D)$

Table 4: Aliased effects in the example 2.2.

in such a type of design with `PLANOR` are:

$$\begin{array}{ll} \text{model} & \texttt{Bl} + (\texttt{A}+\texttt{B}+\texttt{C}+\texttt{D})^2 \\ \text{part to be estimated} & \texttt{A}+\texttt{B}+\texttt{C}+\texttt{D} \end{array} \tag{4}$$

2.2.2 Search by `PLANOR` for several solutions and selection through the study of aliases

It is possible in this small-scale example to request the set of solutions satisfying the requirement defined by (4). A direct syntax to do so is

```
> ABCD3.key <- planor.designkey(factors=c("Bl",LETTERS[1:4]),
+                               nlevels=rep(3,5),
+                               block=~Bl,
+                               model=~Bl+(A+B+C+D)^2,
+                               estimate=~A+B+C+D,
+                               nunits=27,
+                               base=~A+B+C,
+                               max.sol=Inf)

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 3 : key-matrix search
  => search for columns 4 to 5
```

```
      first visit to column 4
      first visit to column 5
The search is closed:  144 solutions found
```

where the `max.sol` argument (maximum number of solutions) has been put at a value larger than necessary[3].

In this way 144 solutions are obtained. Under the hypothesis that interactions of three or more factors are null, one third of these solutions enable the estimation of half of the degrees of freedom for each two-factor interaction. The other two-thirds moreover make the other two degrees of freedom of a given interaction confounded with the blocks. It is not possible to specify to the program that a design of the appropriate third is preferentially sought, but the solutions found can be studied *a posteriori* by the *study of aliases*.

The remainder of this paragraph presents the result of the study of aliases for each type of solution. The solution corresponding to the appropriate third is solution 1 which leaves, for each of the 6 two-factor interactions, 2 unaliased degrees of freedom. The solution corresponding to the inappropriate third is number 34 in which an interaction is completely confounded and for which there are thus only 5 interactions with 2 unaliased degrees of freedom.

The equalities defining each design are presented in the form of a key matrix with one row for each basic factor. Each factor that appears either in the model, in the part to be estimated, in the hierarchies or among the predetermined factors, is associated with a column of this matrix specifying the linear combination of the basic factors which defines this factor.

In fact, several disjoint designs with similar properties are obtained by adding to each factor modulo 3, an integer included between 0 and the "cR" coefficient given at the top of the corresponding column.

The study of aliased effects, also referred to as *alias*, begins with the writing of a synthetic table from which the lists of aliased effects are easily obtained. This table is an intermediate technique which is only of interest for highly skilled users. The ordinary user may, without inconvenience, skip the paragraphs in small type containing this table and related comments.

The standard proposed model for the study of aliases is the model introduced initially for the design search. This may be modified without any problem immediately prior to the study of aliases.

[3]by default, `max.sol=1`; to search for an unlimited number of solutions, `max.sol=Inf` is possible but should be used only for problems of small size; note that it is important to use the `base` argument to avoid redundant solutions

Outputs of the study of aliases for the two solutions adopted

Solution 1 (prime 3)

The levels of a factor at the top of a column are obtained by multiplying the levels of the basic factors appearing at the left of the row by the coefficients in the column and by adding a predetermined integer lower than cR. Calculations of levels performed modulo 3.

		blocs					*
		cR	2	0	2	0	0
			3	3	3	3	3
			A	*B*	*C*	*D*	*Bl*
3	*A*		1	0	0	1	1
3	*B*		0	1	0	1	1
3	*C*		0	0	1	1	0

In R, the design key of Solution 1 is obtained by

```
> print(ABCD3.key[1])

An object of class designkey

********** Prime  3  design **********

  A B C Bl D
A 1 0 0  1 1
B 0 1 0  1 1
C 0 0 1  0 1
```

Study of aliased effects.

The columns of the matrices below provide the defining relations of the design (kernel of the key matrix) from which the aliased effects are inferred.

- The columns of the first matrix generate all the treatment effects confounded with the general mean. By adding the vectors with coordinates $\leq$cT, all the sets of confounded treatment effects can be inferred.
- The linear combinations with coefficients $\leq$cBT in the columns of the second matrix provide the confounded block effects and, for each of these, one of the treatment effects with which it is confounded. The other treatment effects confounded with this block effect are obtained by adding the treatment effects confounded with the mean.
- The sets of treatment effects confounded with each other but not with a block effect can be obtained directly by adding one of the non-zero vectors with coordinates $\leq$cTB to the sets of treatment effects confounded with the blocks.
- The unconfounded block effects are obtained by adding a non-zero vector with coordinates $\leq$cB to the confounded block effects.

- In the sets of confounded effects obtained as described above, a detailed analysis of aliases only includes those aliases which effectively appear in the model (which may lead to differences in the repartition of effects).

							cBT	2
				$order$		3		3
cB	cT	cTB	bl					
0	2	2		3	A	1	A	0
0	2	0		3	B	1	B	0
0	2	2		3	C	1	C	2
0	0	0		3	D	2	D	1
0	0	0	$*$	3	Bl	0	Bl	2
						$\uparrow$		$\uparrow$
				matrix	n^0	1	n^0	2

- List of treatment effects confounded with the mean

$$; ABCD^2; A^2B^2C^2D;$$

- Sets of aliased effects in the model. If the set contains a block effect, the latter is indicated in brackets.

$[Bl]; C^2D; AB;$
$[Bl^2]; CD^2; A^2B^2;$
$AC; B^2D;$
$A^2D; BC;$
$AD^2; B^2C^2;$
$A^2C^2; BD^2;$

- list of unaliased treatment effects

$A; A^2B; B^2; A^2; B; AB^2; C; D; C^2D^2;$
$AD; B^2C; A^2C; BD; C^2; CD; D^2; AC^2;$
$B^2D^2; BC^2; A^2D^2;$

- list of unaliased block effects: empty

In R, this information is obtained by:

```
> alias(ABCD3.key[1], model=~Bl+(A+B+C+D)^2)

********** Prime  3  design **********

UNALIASED TREATMENT EFFECTS
A ; B ; C ; D ; A:B^2 ; A:C^2 ; A^2:D^2 ; B:C^2 ; B:D ; C:D

ALIASED TREATMENT EFFECTS
A:C = B^2:D
A^2:D = B:C
```

```
TREATMENT AND BLOCK EFFECTS CONFOUNDED WITH BLOCK EFFECTS
Bl = A:B = C^2:D

UNALIASED BLOCK EFFECTS
nil
```

Solution 34 **(prime** 3 **)**

Calculations of levels performed modulo 3.

		blocks					$*$
		cR	2	0	2	0	0
			3	3	3	3	3
			A	B	C	D	Bl
3	A		1	0	0	2	2
3	B		0	1	0	1	1
3	C		0	0	1	1	2

Study of aliased effects.

							cBT	2
				ordre		3		3
cB	cT	cTB	bl					
0	2	2		3	A	1	A	0
0	2	2		3	B	2	B	0
0	2	0		3	C	2	C	2
0	0	0		3	D	1	D	2
0	0	0	$*$	3	Bl	0	Bl	1

- List of treatment effects confounded with the mean

$$; AB^2C^2D; A^2BCD^2;$$

- Sets of aliased effects in the model. If the set contains a block effect, the latter is indicated in brackets.

$$[Bl^2]; C^2D^2;$$
$$[Bl]; CD;$$
$$A^2C; B^2D;$$
$$A^2D^2; B^2C^2;$$
$$A^2B; C^2D;$$
$$AB^2; CD^2;$$
$$BC; AD;$$
$$BD^2; AC^2;$$

- list of unaliased treatment effects

$$A; BD; BC^2; A^2; B^2C; B^2D^2; B; AC;$$
$$AD^2; AB; C; D^2; B^2; A^2D; A^2C^2; D; C^2; A^2B^2;$$

- list of unaliased block effects

In R, the information is given by

```
> alias(ABCD3.key[34], model=~Bl+(A+B+C+D)^2)

********** Prime  3  design **********

UNALIASED TREATMENT EFFECTS
A ; B ; C ; D^2 ; A:B ; A^2:D ; B:C^2 ; B^2:D^2 ; C^2:D^2

ALIASED TREATMENT EFFECTS
A:B^2 = C:D^2
A:C^2 = B:D^2
A:D = B:C

TREATMENT AND BLOCK EFFECTS CONFOUNDED WITH BLOCK EFFECTS
Bl = A:C

UNALIASED BLOCK EFFECTS
nil
```

2.2.3 Randomization of a block design

Randomization takes place in two stages. In the first stage, the numbers associated with the real blocks are associated randomly to each block number appearing in the systematic design. The unit number allocated to each treatment within each block is then drawn at random. Table 5 shows a possible result of this randomization which leads to the columns entitled Bl_0 and *ind-rep* of table 3. Column Bl_0 provides the number of the real block, and *ind-rep* provides that of the block unit. In the second stage, several related transformations are required performed in order to obtain a ““ready-for-use” design, in particular, sorting of Bl_0 and *ind-rep*.

2.3 Example of a combination of factors with 6, 4 and 2 levels

2.3.1 Definition and properties. Decomposition into pseudofactors

There are 144 existing units distributed into 6 blocks – factor Bl –, over which 4 treatment factors A, B, C, D with 6, 6, 4 and 2 levels, respectively, are tested. Apart from D which

block nb. in the systematic design	0	1	2
real block nb.	2	1	0

Bl	nb. of the units allocated to block treatments								
0	2	1	5	7	0	4	8	6	3
1	6	0	2	8	7	4	5	3	1
2	6	1	7	8	2	4	3	0	5

Table 5: Randomization of a block design

only has 2 levels, each of these factors is decomposed into 2 *pseudofactors* as indicated in table 6. Note that the program uses the symbol "_" to indicate subscripting: A_1 for A_1, B_2 for B_2, etc ...

A	A_1	A_2	B	B_1	B_2	Bl	Bl_1	Bl_2	C	C_1	C_2
1	1	2	1	1	2	1	1	2	1	1	1
2	1	1	2	1	1	2	1	1	2	1	0
3	1	0	3	1	0	3	1	0	3	0	1
4	0	2	4	0	2	4	0	2	4	0	0
5	0	1	5	0	1	5	0	1			
6	0	0	6	0	0	6	0	0			

Table 6: Decomposition into pseudofactors in example 2.3.

Factors A, B, C are taken as basic factors. Then Bl and D are defined by the following equalities:

$$\begin{array}{lcll} Bl_1 & = & A_1 + B_1 + C_1 & (\text{mod } 2) \\ Bl_2 & = & A_2 + 2B_2 & (\text{mod } 3) \\ D & = & A_1 + B_1 + C_1 + C_2 & (\text{mod } 2) \end{array} \qquad (5)$$

2.3.2 Study of aliases (not updated subsection)

In order to perform the study of aliases for this specific design, the equalities defined by (5) are inserted in the following form:

$$\begin{array}{lcl} Bl\_1 & : & A\_1 + B\_1 + C\_1 \\ Bl\_2 & : & A\_2 + 2B\_2 \\ D & : & A\_1 + B\_1 + C\_1 + C\_2 \end{array}$$

We specify in the frame *added factors* of this same screen that Bl is a block factor.

It is assumed that the model contains a block effect and a treatment effect including the main effects and interactions of two factors. In view of the fact that it is automatically completed by `PLANOR` , this model may be written as:

$$Bl + A.B + A.C + A.D + B.C + B.D + C.D \ ,$$

or even using a model part

$$\begin{array}{ll} \text{model part} & P : A + B + C + D \\ \text{model} & Bl + P.P \end{array}$$

On the basis of this model entered in screen 6 (see § 5.6 and figure **??**), the study of aliases provides the results which follow. In the current version, these results are given separately for primes 2 and 3 which divide the number of units.

Prime 2

- List of treatment effects confounded with the mean

$$; A_1B_1C_1C_2D;$$

- Sets of confounded effects in the model. When the set contains a block effect, the latter is indicated in brackets.

$$\begin{array}{l} [Bl_1]; C_2D; \\ A_1B_1; C_1C_2D; \\ A_1D; B_1C_1C_2; \\ B_1D; A_1C_1C_2; \end{array}$$

- list of unaliased treatment effects

$$A_1; B_1C_1; B_1; A_1C_1; C_1; C_2; D; A_1C_2; B_1C_2; C_1D; C_1C_2;$$

- list of unaliased block effects: empty.

Prime 3

- List of treatment effects confounded with the mean: empty.
- Sets of aliased effects in the model. If the set has a block effect, the latter is indicated in brackets.

$$\begin{array}{l} [Bl_2^2]; A_2^2B_2; \\ [Bl_2]; A_2B_2^2; \end{array}$$

- list of unaliased treatment effects

$$; A_2; B_2; A_2^2B_2^2; A_2^2; A_2B_2; B_2^2;$$

- list of unaliased block effects: empty

The aliased effects in the global design can easily be deduced from these studies conducted separately for primes 2 and 3. The general rule is simple. Every effect is decomposed into a product $\alpha_2\alpha_3$ in which α_2 and α_3 may be expressed from the 2- and 3-level pseudofactors, respectively. For example, $\alpha_2 = A_1D$ and $\alpha_3 = A_2B_2^2$ for the effect $A_1A_2B_2^2D$. So $\alpha_2\alpha_3$ and $\beta_2\beta_3$ are two effects thus decomposed.

Rule 2 *$\alpha_2\alpha_3$ and $\beta_2\beta_3$ are confounded if and only if α_2 is confounded with β_2 and α_3 is confounded with β_3.*

In particular, if neither α_2, nor α_3 are confounded, the effect $\alpha_2\alpha_3$ is not confounded. For instance, in this example A_1, A_2 et A_2^2 are not confounded, and neither are the products A_1A_2, $A_1A_2^2$. The 5 degrees of freedom of A are thus estimable. Similarly, it can be observed that the other main effects are estimable in the model which includes all the interactions of two treatment factors and the block effect.

Let us now examine an interaction such as AB. The 25 corresponding effects are all shown in table 7 with, in the same cell, the other effects with which they are confounded. Only three degrees of freedom are confounded with non-negligible interactions?: A_1B_1 confounded with C_1C_2D, $A_2B_2^2$, $A_2^2B_2$ confounded with Bl_2, Bl_2^2.

	A_1	B_1	A_1B_1	C_1C_2D
A_2	A_1A_2	B_1A_2	$A_1B_1A_2$	$A_2C_1C_2D$
A_2^2	$A_1A_2^2$	$B_1A_2^2$	$A_1B_1A_2^2$	$A_2^2C_1C_2D$
B_2	A_1B_2	B_1B_2	$A_1B_1B_2$	$B_2C_1C_2D$
B_2^2	$A_1B_2^2$	$B_1B_2^2$	$A_1B_1B_2^2$	$B_2^2C_1C_2D$
A_2B_2	$A_1A_2B_2$	$B_1A_2B_2$	$A_1B_1A_2B_2$	$A_2B_2C_1C_2D$
$A_2^2B_2^2$	$A_1A_2^2B_2^2$	$B_1A_2^2B_2^2$	$A_1B_1A_2^2B_2^2$	$A_2^2B_2^2C_1C_2D$
$A_2B_2^2$	$A_1A_2B_2^2$	$B_1A_2B_2^2$	$A_1B_1A_2B_2^2$	$A_2B_2^2C_1C_2D$
Bl_2	A_1Bl_2	B_1Bl_2	$A_1B_1Bl_2$	$C_1C_2DBl_2$
$A_2^2B_2$	$A_1A_2^2B_2$	$B_1A_2^2B_2$	$A_1B_1A_2^2B_2$	$A_2^2B_2C_1C_2D$
Bl_2^2	$A_1Bl_2^2$	$B_1Bl_2^2$	$A_1B_1Bl_2^2$	$C_1C_2DBl_2^2$

Table 7: Confounding for effects involving 2- and 3-level pseudofactors

3 Examples of designs for the ARILAIT robot

With a view on measuring the effectiveness of cleaning and disinfection of open surfaces, the LGHPA laboratory and the ARILAIT association, already mentioned in the introduction, set up a robot which could soil, then clean and disinfect a surface in reproducible conditions [6]. The surface is a stainless steel plate containing 16 circular test specimens measuring 5 cm in diameter, laid out on the plate as indicated in figure 1.

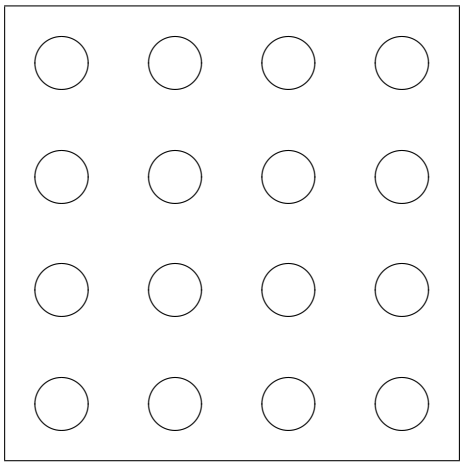


Figure 1: Position of test specimens on the plate

A variety of factors are likely to influence effectiveness of cleaning and disinfection. These include, for instance, the nature of the material of which the test specimen is made, its roughness, the type and quantity of soiling, bacterial concentration in the soiled area, the nature, concentration, application temperature and duration of action of each of the products used for cleaning and disinfection.

Operating the robot is subject to certain constraints. It moves by row and column, and the levels of factors cannot vary ad libitum from one specimen to the other. Furthermore, effectiveness of cleaning and disinfection is assessed by comparing the cleaned and disinfected specimen with a control specimen that is similarly soiled but not cleaned. For each treatment, at least at the soiling stage, two identically treated specimens are required.

3.1 Creation of a full factorial design that incorporates manipulation constraints

3.1.1 Objective and constraints

In the example in this paragraph, a single commercial product - a chlorinated alkaline agent - is applied in the form of foam to clean and disinfect. The factors studied are

n-soil :	nature of the soiling	curd Saint-Paulin.
q-soil :	quantity of soiling deposited on the test specimens, varied by placing a weight on the robot arm	?0.01 g/specimen 0.10 g/specimen.
rough :	roughness of test specimens	0.25μm 0.73μm.
conc :	concentration of the cleaning and disinfection product	1% 3% (v/v)
T-act :	duration of action of the product	15mn 30mn.

In this experiment, there is no watertight partition preventing the diffusion of foam from one part of the plate to another. The latter two factors, concentration and duration of action of the product thus necessarily remain constant throughout the same manipulation of the plate. In order to study the 4 combinations of these two factors, (1%, 15mn), (1%, 30mn), (3%, 15mn), (3%, 30mn), at least 4 such manipulations are required, that is to say, 4 plates for 4 weeks of experimenting. To study variability from one plate to another, two of these combinations, (3%, 15mn) and (1%, 30mn), are repeated, which results in a total of 6 plates.

The 8 combinations of levels of the other three factors can be tested on each of the 6 plates: (curd , 0.01g/e, 0.25μm), (curd , 0.10g/e, 0.25μm), (curd , 0.01g/e, 0.73μm), (curd , 0.10g/e, 0.73μm), (StPaul, 0.01g/e, 0.25μm), (StPaul, 0.10g/e, 0.25μm), (StPaul, 0.01g/e, 0.73μm), (StPaul, 0.10g/e, 0.73μm), by placing a treated test specimen and its associated control for each of these treatments.

To facilitate manipulations, soiling is performed column by column, and the nature and quantity of soiling are only modified when moving from one column to another. Two test specimens of each roughness, with one acting as a control for the other, are placed in each column. At the cleaning stage, the treated specimens are replaced in the same position and the control specimens are replaced by other specimens which are only used to fill the holes.

In this example, the design is thus clearly defined and an algorithmic search is useless. Nevertheless, it is worth considering how to proceed to create and randomize this design practically.

3.1.2 Creation by juxtaposition of two sub-designs

As already mentioned, the 6 plates are made up of a 4-plate design including the set of 4×8 treatments and of a 2-plate design including half of these treatments.

These two designs, denoted by ROBOT1A and ROBOT1B, respectively, are obtained separately and then merged. For each design, the basic factors are the plate number *plate*, the column number in the plate *col*, the unit number in the column *u*. The defined factors are the treatment factors: *n-soil, q-soil, rough, conc, T-act*. Note that on

this example, the order of the factors has a large influence on the computing time for the design search to be run below. Putting factors *conc* and *Tact* after factor *rough* results in a much larger computing time because the solutions are then farther in the search tree.

```
> robotFac <- planor.factors( factors=list(
+                          conc=c("1%","3%"),
+                          Tact=c("15mn","30mn"),
+                          nsoil=c("curd","Saint-Paulin"),
+                          qsoil=c("10mg","100mg"),
+                          rough=c(0.25,0.73),
+                          plate=1:4,
+                          col=1:4,
+                          u=1:2),
+                        block=~plate+col,
+                        hierarchy=list(~nsoil/(plate*col),
+                                       ~qsoil/(plate*col),
+                                       ~conc/plate,
+                                       ~Tact/plate))
```

The "hierarchy" argument specifies that *conc* and *T-act* must be constant on each plate and *n-soil* and *q-soil* must be constant on each column in each plate. It follows that in the designs, *conc* and *T-act* will have to be defined from the plate number *plate*, and *n-soil*, *q-soil* from the plate number *pl* and the columnn number in the plate *col*.

The ROBOT1A design must be complete, *i.e.*, involve each of the 2^5 treatments defined by *n-soil, q-soil, rough, conc, T-act*. In order to ensure this, a model including all main effects and interactions of the 5 factors is specified. In addition, the *rough* factor must be orthogonal to the block factors *plate, col*. This is specified in the `listofmodels` argument of.`planor.model`.

```
> robotMod <- planor.model( model=~nsoil * qsoil * rough * conc * Tact,
+                       listofmodels=list(c(~plate*col+rough,~rough)) )
```

The designkey can now be searched.

```
> robotKeyA <- planor.designkey(factors=robotFac,
+                               model=robotMod,
+                               nunits=32,
+                               base=~plate+col+u)


Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 6 to 10
      first visit to column 6
      first visit to column 7
```

```
      first visit to column 8
      first visit to column 9
      first visit to column 10
The search is closed: max.sol =  1 solution(s) found
```

The ROBOT1A instructions may be used as the starting point for the construction of the ROBOT1B design. In `planor.factors`, the levels of *plate* are replaced by 5 and 6 and in `planor.model`, *T-act*, for instance, is deleted from the model. However, *T-act* does not appear in the model, and there is nothing to preclude it from remaining constant in this second design. In order to compel it to assume both its levels, a second model is developed, containing only *T-act* together with an empty part to be estimated.

```
> robotFacB <- planor.factors(factors=list(
+                          nsoil=c("curd","Saint-Paulin"),
+                          qsoil=c("10mg","100mg"),
+                          rough=c(0.25,0.73),
+                          conc=c("1%","3%"),
+                          Tact=c("15mn","30mn"),
+                          plate=5:6,
+                          col=1:4,
+                          u=1:2),
+                       block=~plate+col,
+                       hierarchy=list(~nsoil/(plate*col),
+                                      ~qsoil/(plate*col),
+                                      ~conc/plate,
+                                      ~Tact/plate))
> robotModB <- planor.model( model=~nsoil * qsoil * rough * conc,
+                       listofmodels=list(c(~plate*col+rough,~rough),
+                                         c(~Tact,~Tact)) )
> robotKeyB <- planor.designkey(factors=robotFacB, model=robotModB,
+                       nunits=16, base=~plate+col+u )

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 5 to 9
      first visit to column 5
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
The search is closed: max.sol =  1 solution(s) found
```

The results of the `PLANOR` search are saved in `robotKeyA` and `robotKeyB`. They contain the values of the parameters determining the search and the key matrices.

The design keys (Table 8) can be printed by

robotKeyA

```
An object of class listofkeyrings

********** Prime  2  design **********

--- Solution  1  for prime  2  ---

        plate_1 plate_2 col_1 col_2 u conc Tact nsoil qsoil rough
plate_1       1       0     0     0 0    1    0     0     0     0
plate_2       0       1     0     0 0    0    1     0     0     0
col_1         0       0     1     0 0    0    0     1     0     0
col_2         0       0     0     1 0    0    0     0     1     0
u             0       0     0     0 1    0    0     0     0     1
```

robotKeyB

```
An object of class listofkeyrings

********** Prime  2  design **********

--- Solution  1  for prime  2  ---

      plate col_1 col_2 u nsoil qsoil rough conc Tact
plate     1     0     0 0     0     0     0    1    1
col_1     0     1     0 0     1     0     0    0    0
col_2     0     0     1 0     0     1     0    0    0
u         0     0     0 1     0     0     1    0    0
```

Table 8: Key matrices of the ROBOT1A and ROBOT1B designs

```
> print(robotKeyA)
> print(robotKeyB)
```

Here, these matrices lead to taking *n-soil*=col_1 , *q-soil*=col_2, *rough*=*row, conc*=$plate_1$, *T-act*=$plate_2$ to construct ROBOT1A and *n-soil*=col_1 , *q-soil*=col_2, *rough*=col_1 + *row, conc*=*plate, T-act*=*plate* for ROBOT2. col_1, col_2, denoted by *col*_1, *col*_2 in the program, are the pseudofactors resulting from the decomposition of the 4-level factor *col* and, similarly, $plate_1$, $plate_2$, denoted by *plate*_1, *plate*_2, are those resulting from the decomposition of factor *plate* in the ROBOT1A design.

By default, a single solution is requested. In this case, after obtaining the single key matrix in cases A and B, the design can be constructed by binding the designs generated by the `planor.design` function. By precaution, the column order is specified explicitly when binding the two dataframes.

```
> robotPlanA <- planor.design(robotKeyA)@design
> robotPlanB <- planor.design(robotKeyB)@design
> robotPlan <- rbind(robotPlanA[c("plate","col",
+                                 "nsoil","qsoil",
+                                 "rough","conc","Tact")],
+                    robotPlanB[c("plate","col",
```

```
+                                    "nsoil","qsoil",
+                                    "rough","conc","Tact")])
```

The designs (Table 9) can be printed by

```
> print(robotPlanA)
> print(robotPlanB)
> print(robotPlan)
```

3.1.3 Randomization taking into account block structure

In order to be effectively implemented, this ROBOT1 design must be randomized by the function `planor.randomize`. The randomization process must involve switching plates around so that concentration and duration of action are always constant for each plate following randomization. Similarly, the columns of each plate must be switched around. This is indicated to the program by the randomization model which includes the terms `plate`, `plate:col`, and `plate:col:UNITS`.

```
> robotRand <- planor.randomize(~plate/col/UNITS, data=robotPlan)
> write.table(robotRand, file="ROBOT.out", quote=FALSE,
+                         row.names=FALSE, sep=" ")
```

The random permutation of the 6 plates and the 6 random permutations of the 4 columns of each plate are completed by the 24 permutations of the 2 units in each column. The randomized design units are reordered appropriately for manipulation by the robot. The randomized design is stored in the `robotRand` object and written in the ASCII file ROBOT.out.

The randomized design `robotRand` can be printed by `print(robotRand)` (Table 10).

Remarks:

- The fact that *col* only appears in the randomization model in conjunction with *plate* indicates a hierarchical relationship and entails that random permutations of column numbers occur independently from one plate to the other. Similarly, the repetition index, considered to be nested within all the other factors, is separately randomized in each column of each plate. A different choice of model could be made if the plate was in a vertical position with, for example, column 1 at the top and column 4 at the bottom. It would then be necessary to take into account, both in the construction and in the randomization of the design, a potential column effect arising from differences in soiling or cleaning as a function of height. The column factor would then be crossed with – and no longer nested within – the plate factor.

robotPlanA

	plate	col	u	conc	Tact	nsoil	qsoil	rough
1	1	1	1	1%	15mn	curd	10mg	0.25
2	1	1	2	1%	15mn	curd	10mg	0.73
3	1	2	1	1%	15mn	curd	100mg	0.25
4	1	2	2	1%	15mn	curd	100mg	0.73
5	1	3	1	1%	15mn	Saint-Paulin	10mg	0.25
6	1	3	2	1%	15mn	Saint-Paulin	10mg	0.73
7	1	4	1	1%	15mn	Saint-Paulin	100mg	0.25
8	1	4	2	1%	15mn	Saint-Paulin	100mg	0.73
9	2	1	1	1%	30mn	curd	10mg	0.25
10	2	1	2	1%	30mn	curd	10mg	0.73
11	2	2	1	1%	30mn	curd	100mg	0.25
12	2	2	2	1%	30mn	curd	100mg	0.73
13	2	3	1	1%	30mn	Saint-Paulin	10mg	0.25
14	2	3	2	1%	30mn	Saint-Paulin	10mg	0.73
15	2	4	1	1%	30mn	Saint-Paulin	100mg	0.25
16	2	4	2	1%	30mn	Saint-Paulin	100mg	0.73
17	3	1	1	3%	15mn	curd	10mg	0.25
18	3	1	2	3%	15mn	curd	10mg	0.73
19	3	2	1	3%	15mn	curd	100mg	0.25
20	3	2	2	3%	15mn	curd	100mg	0.73
21	3	3	1	3%	15mn	Saint-Paulin	10mg	0.25
22	3	3	2	3%	15mn	Saint-Paulin	10mg	0.73
23	3	4	1	3%	15mn	Saint-Paulin	100mg	0.25
24	3	4	2	3%	15mn	Saint-Paulin	100mg	0.73
25	4	1	1	3%	30mn	curd	10mg	0.25
26	4	1	2	3%	30mn	curd	10mg	0.73
27	4	2	1	3%	30mn	curd	100mg	0.25
28	4	2	2	3%	30mn	curd	100mg	0.73
29	4	3	1	3%	30mn	Saint-Paulin	10mg	0.25
30	4	3	2	3%	30mn	Saint-Paulin	10mg	0.73
31	4	4	1	3%	30mn	Saint-Paulin	100mg	0.25
32	4	4	2	3%	30mn	Saint-Paulin	100mg	0.73

robotPlanB

	plate	col	u	nsoil	qsoil	rough	conc	Tact
1	5	1	1	curd	10mg	0.25	1%	15mn
2	5	1	2	curd	10mg	0.73	1%	15mn
3	5	2	1	curd	100mg	0.25	1%	15mn
4	5	2	2	curd	100mg	0.73	1%	15mn
5	5	3	1	Saint-Paulin	10mg	0.25	1%	15mn
6	5	3	2	Saint-Paulin	10mg	0.73	1%	15mn
7	5	4	1	Saint-Paulin	100mg	0.25	1%	15mn
8	5	4	2	Saint-Paulin	100mg	0.73	1%	15mn
9	6	1	1	curd	10mg	0.25	3%	30mn
10	6	1	2	curd	10mg	0.73	3%	30mn
11	6	2	1	curd	100mg	0.25	3%	30mn
12	6	2	2	curd	100mg	0.73	3%	30mn
13	6	3	1	Saint-Paulin	10mg	0.25	3%	30mn
14	6	3	2	Saint-Paulin	10mg	0.73	3%	30mn
15	6	4	1	Saint-Paulin	100mg	0.25	3%	30mn
16	6	4	2	Saint-Paulin	100mg	0.73	3%	30mn

robotPlan

	plate	col	nsoil	qsoil	rough	conc	Tact
1	1	1	curd	10mg	0.25	1%	15mn
2	1	1	curd	10mg	0.73	1%	15mn
3	1	2	curd	100mg	0.25	1%	15mn
4	1	2	curd	100mg	0.73	1%	15mn
5	1	3	Saint-Paulin	10mg	0.25	1%	15mn
6	1	3	Saint-Paulin	10mg	0.73	1%	15mn
7	1	4	Saint-Paulin	100mg	0.25	1%	15mn
8	1	4	Saint-Paulin	100mg	0.73	1%	15mn
9	2	1	curd	10mg	0.25	1%	30mn
10	2	1	curd	10mg	0.73	1%	30mn
11	2	2	curd	100mg	0.25	1%	30mn
12	2	2	curd	100mg	0.73	1%	30mn
13	2	3	Saint-Paulin	10mg	0.25	1%	30mn
14	2	3	Saint-Paulin	10mg	0.73	1%	30mn
15	2	4	Saint-Paulin	100mg	0.25	1%	30mn
16	2	4	Saint-Paulin	100mg	0.73	1%	30mn
17	3	1	curd	10mg	0.25	3%	15mn
18	3	1	curd	10mg	0.73	3%	15mn
19	3	2	curd	100mg	0.25	3%	15mn
20	3	2	curd	100mg	0.73	3%	15mn
21	3	3	Saint-Paulin	10mg	0.25	3%	15mn
22	3	3	Saint-Paulin	10mg	0.73	3%	15mn
23	3	4	Saint-Paulin	100mg	0.25	3%	15mn
24	3	4	Saint-Paulin	100mg	0.73	3%	15mn
25	4	1	curd	10mg	0.25	3%	30mn
26	4	1	curd	10mg	0.73	3%	30mn
27	4	2	curd	100mg	0.25	3%	30mn
28	4	2	curd	100mg	0.73	3%	30mn
29	4	3	Saint-Paulin	10mg	0.25	3%	30mn
30	4	3	Saint-Paulin	10mg	0.73	3%	30mn
31	4	4	Saint-Paulin	100mg	0.25	3%	30mn
32	4	4	Saint-Paulin	100mg	0.73	3%	30mn
33	5	1	curd	10mg	0.25	1%	15mn
34	5	1	curd	10mg	0.73	1%	15mn
35	5	2	curd	100mg	0.25	1%	15mn
36	5	2	curd	100mg	0.73	1%	15mn
37	5	3	Saint-Paulin	10mg	0.25	1%	15mn
38	5	3	Saint-Paulin	10mg	0.73	1%	15mn
39	5	4	Saint-Paulin	100mg	0.25	1%	15mn
40	5	4	Saint-Paulin	100mg	0.73	1%	15mn
41	6	1	curd	10mg	0.25	3%	30mn
42	6	1	curd	10mg	0.73	3%	30mn
43	6	2	curd	100mg	0.25	3%	30mn
44	6	2	curd	100mg	0.73	3%	30mn
45	6	3	Saint-Paulin	10mg	0.25	3%	30mn
46	6	3	Saint-Paulin	10mg	0.73	3%	30mn
47	6	4	Saint-Paulin	100mg	0.25	3%	30mn
48	6	4	Saint-Paulin	100mg	0.73	3%	30mn

Table 9: The three designs robotPlanA, robotPlanB, and robotPlan before randomization

	plate	col	nsoil	qsoil	rough	conc	Tact
1	1	1	curd	10mg	0.25	1%	30mn
2	1	1	curd	10mg	0.73	1%	30mn
3	1	2	Saint-Paulin	10mg	0.25	1%	30mn
4	1	2	Saint-Paulin	10mg	0.73	1%	30mn
5	1	3	Saint-Paulin	100mg	0.73	1%	30mn
6	1	3	Saint-Paulin	100mg	0.25	1%	30mn
7	1	4	curd	100mg	0.73	1%	30mn
8	1	4	curd	100mg	0.25	1%	30mn
9	2	1	Saint-Paulin	10mg	0.73	3%	15mn
10	2	1	Saint-Paulin	10mg	0.25	3%	15mn
11	2	2	curd	10mg	0.25	3%	15mn
12	2	2	curd	10mg	0.73	3%	15mn
13	2	3	curd	100mg	0.73	3%	15mn
14	2	3	curd	100mg	0.25	3%	15mn
15	2	4	Saint-Paulin	100mg	0.73	3%	15mn
16	2	4	Saint-Paulin	100mg	0.25	3%	15mn
17	3	1	Saint-Paulin	100mg	0.73	3%	30mn
18	3	1	Saint-Paulin	100mg	0.25	3%	30mn
19	3	2	curd	10mg	0.73	3%	30mn
20	3	2	curd	10mg	0.25	3%	30mn
21	3	3	Saint-Paulin	10mg	0.73	3%	30mn
22	3	3	Saint-Paulin	10mg	0.25	3%	30mn
23	3	4	curd	100mg	0.73	3%	30mn
24	3	4	curd	100mg	0.25	3%	30mn
25	4	1	curd	10mg	0.25	1%	15mn
26	4	1	curd	10mg	0.73	1%	15mn
27	4	2	curd	100mg	0.25	1%	15mn
28	4	2	curd	100mg	0.73	1%	15mn
29	4	3	Saint-Paulin	100mg	0.25	1%	15mn
30	4	3	Saint-Paulin	100mg	0.73	1%	15mn
31	4	4	Saint-Paulin	10mg	0.25	1%	15mn
32	4	4	Saint-Paulin	10mg	0.73	1%	15mn
33	5	1	curd	100mg	0.73	3%	30mn
34	5	1	curd	100mg	0.25	3%	30mn
35	5	2	Saint-Paulin	100mg	0.73	3%	30mn
36	5	2	Saint-Paulin	100mg	0.25	3%	30mn
37	5	3	Saint-Paulin	10mg	0.25	3%	30mn
38	5	3	Saint-Paulin	10mg	0.73	3%	30mn
39	5	4	curd	10mg	0.25	3%	30mn
40	5	4	curd	10mg	0.73	3%	30mn
41	6	1	Saint-Paulin	100mg	0.73	1%	15mn
42	6	1	Saint-Paulin	100mg	0.25	1%	15mn
43	6	2	curd	10mg	0.73	1%	15mn
44	6	2	curd	10mg	0.25	1%	15mn
45	6	3	curd	100mg	0.73	1%	15mn
46	6	3	curd	100mg	0.25	1%	15mn
47	6	4	Saint-Paulin	10mg	0.25	1%	15mn
48	6	4	Saint-Paulin	10mg	0.73	1%	15mn

Table 10: Randomized ROBOT1 design (after automatic sorting)

- The type of randomization performed, thoroughly studied in [4], normally leads to a model for analysis in which a certain number of terms appear, referred to as *ancestral terms*, which are automatically deduced from the randomization model. In this example, the terms appearing in the ASCII file created by randomization, are *plate*, *plate.col*, *plate.col.UNITS*. Associated effects in the variance analysis model are random and the analysis is normally performed by a procedure that takes into account these random effects (see [3], [20]).

- The introduction of factors *plate* and *col* is rendered indispensable here by the robot manipulation constraints. Nevertheless, the plate effect was shown to be negligible during the first trials, undoubtedly owing to the coupling of each treated specimen with a control specimen, and it is natural to think that the column effect, on the horizontally maintained plate, is even more negligible. In other words, the excellent reproducibility of the operations performed by the robot legitimately leads to the assumption that the variability between two different column units is similar to that between the units of the same column. In these circumstances, it is permissible not to take into account the column effect during the analysis.

3.2 Designs for a plate

With a view to implementing the designs described below, removable watertight partitions are now placed to separate either the columns or rows of the plate. Subsequently, treatments may be adjusted from one column to the next, at each stage of the experiment.

Moreover, soiling now takes place in two steps. At the end of each step, 8 *active* specimens are obtained together with the 8 associated *control* specimens. In the following stages of the experiment, cleaning and disinfection, the $16 = 2 \times 8$ active specimens are laid out simultaneously on the 16 parts of the plate.

With 16 units, it is possible to study up to five 2-level factors in resolution 5, and up to 8 in resolution 4. For an intermediate number of factors such as 6 or 7, it is impossible to achieve a resolution better than 4. More specifically, if the model includes all the main effects and interactions of two factors, it is impossible to estimate, in addition to the main effects, even a single 2-factor interaction.

It might be inferred from this that it is always preferable to use 8 rather than 7 or 6 factors, but that is incorrect since the size of confounded sets of interactions increases with the number of factors (table 11). And if the test relating to one of these sets is significant, the interpretation is therefore more intricate if the number of factors is higher (see [20] for an example of interpretation of this type of design).

Furthermore, if certain interactions are assumed to be negligible, or if the estimation of certain main effects which are already known is not required, it becomes possible to estimate certain interactions. In this case, the number of factors in the design, 6, 7 or 8, can have an essential impact.

The above considerations reveal the type of set-up which can be sought to implement when conducting the experiment with only one plate. In practice, it is necessary to take

Basic factors	Defined factors	
A, B, C, D	$E = ABC$, $F = ABD$, $G = ACD$, $H = BCD$	
Confounding between interactions		
with the 6 factors	with the 7 factors	with the 8 factors
A, B, C, D, E, F	A, B, C, D, E, F, G	A, B, C, D, E, F, G, H
AB; CE; DF	AB; DF; CE	AB; DF; GH; CE
AC; BE	AC; DG; BE	AC; FH; DG; BE
BC; AE	BC; FG; AE	BC; DH; FG; AE
AD; BF	AD; CG; BF	AD; EH; CG; BF
BD; AF	BD; AF; EG	BD; EG; AF; CH
CD; EF	CD; AG; EF	CD; EF; AG; BH
DE; CF	BG; CF; DE	BG; CF; AH; DE

Table 11: Designs of resolution 4 for 16 units and 6 to 8 factors

into account the constraints imposed by robot manipulation. However, the following two examples show that if the column and row effects are neglected, adaptation to constraints may be achieved without any additional reduction of the number of factors studied.

3.2.1 Design of resolution 4 for 8 factors

In this example, there are eight treatment factors at two levels : *n-soil*, *q-soil*, *c-bact*, *T-act*, *conc*, *brush*, *rough*, *nat*. The unique plate is divided into two macro-columns of two columns (factors *col1, col2*) and two macro-rows of two rows (*row1, row2*) as indicated in Figure 2.

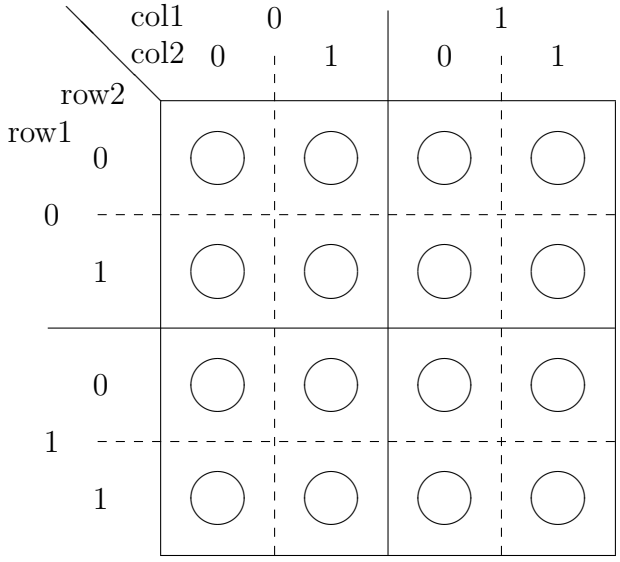


Figure 2: Definition of block systems (*col1, col2, row1, row2*)

In each macro-row, at the *first soiling*, the active specimens are laid out along one of the rows (*row2*= 0) while the controls are laid out along the other (*row2*= 1). The roles of the two rows are switched at the *second soiling* (*row2*= 0: controls, *row2*= 1: active

test specimens). In the subsequent stages, each of the active specimens is replaced in the position occupied during soiling.

To facilitate manipulations, soiling is performed column by column and cleaning is performed row by row. Soiling thus cannot vary within the same column. It follows that the nature of the soiling (*n-soil*) and its bacterial concentration (*c-bact*) must be identical for the two active specimens soiled simultaneously. Consequently, the levels of both factors must be defined from the pseudofactor levels *col1, col2, row2*. This is indicated in the first two hierarchy formulae specified in the `hierarchy` argument of `planor.factors` below.

Similarly, concentration (*conc*) and duration of action (*T-act*) of the cleaning product can only be modified between rows. The levels of these factors are defined from *row1* and *row2*, which is also indicated in the hierarchy list.

In this case, the possibility is recognized that the weight on the robot arm may be changed in the middle of the column, to vary the quantity of soiling *q-soil*. Therefore, no hierarchical constraint is required to be taken into account for this factor.

The intensity of brushing *brush* is equally modified by adding a weight on the robot arm. To avoid too frequent changes, this modification is only authorized every two units, that is to say, only in the middle and at the end of each row. In order to take this constraint into account, the last row of the hierarchy field is introduced which makes the level of *brush* dependent on the row (*row1, row2*) and the macro-column (*col1*).

```
> robpl1r4.fac <- planor.factors(factors=list(
+                                  row1=1:2,
+                                  row2=1:2,
+                                  col1=1:2,
+                                  col2=1:2,
+                                  nsoil=c("curd","Saint-Paulin"),
+                                  qsoil=c("10mg","100mg"),
+                                  cbact=c("3%","6%"),
+                                  Tact=c("15mn","30mn"),
+                                  conc=c("1%","3%"),
+                                  brush=c("strong","weak"),
+                                  rough=c(0.25,0.75),
+                                  nat=1:2),
+                                block=~row1+row2+col1+col2,
+                                hierarchy=list(~nsoil/(col1*col2*row2),
+                                  ~cbact/(col1*col2*row2),
+                                  ~Tact/(row1*row2),
+                                  ~conc/(row1*row2),
+                                  ~brush/(row1*row2*col1)))
> robpl1r4.mod <- planor.model( listofmodels=list(
+   c( ~row2 + (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2,
+      ~nsoil+qsoil+cbact+Tact+conc+brush+rough+nat),
+   c( ~col1*col2*row2 + row1*row2*col1,
+      ~rough+nat)) )
```

```
> robpl1r4.key <- planor.designkey(factors=robpl1r4.fac,
+                                  model=robpl1r4.mod,
+                                  nunits=16,
+                                  base=~row1+row2+col1+col2 )

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 5 to 12
      first visit to column 5
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
      first visit to column 10
      first visit to column 11
      first visit to column 12
The search is closed: max.sol =  1 solution(s) found

>
> #summary(robpl1r4.key)
```

The first model specified in `planor.model` contains all the main effects and two-factor interactions involving the 8 treatment factors, in addition to *row2*. The part to be estimated contains all the main effects. The resulting design is thus of resolution 4. Moreover, the inclusion of factor *row2* relating to the soiling number ensures that all the main treatment effects are estimable, even in the presence of an additive soiling effect.

In contrast, the other column or row factors *col1, col2, row1* are not taken into account in the first model, for the simple reason that the hierarchical constraints do not enable this. Thus if the model includes the term *row1.row2*, there can be no solution since *T-act* and *conc* are constant in each row.

In the framework of a model including row and column effects, it is thus impossible to estimate all the main effects. Nevertheless, it can be ensured that some of these effects are estimable. For instance in this example, adding the second pair *model—part to be estimated* ensures that the main effects of roughness and "nature of the material making up the specimen" (*rough, nat*) are estimable, even if the model includes, in addition to the row and column effects, the interactions *column×soiling (col1.col2.row2)* and *macrocolumn×row (col1.row1.row2)*. In the resulting set-up, two active specimens located in the same half-row differ both in terms of roughness and nature, and this also applies to two active specimens that are soiled simultaneously in the same column.

Randomization of the design must be consistent with the row and column structure and with the macro-column and macro-row structure, to avoid inappropriate weight changes on the robot arm, during soiling or cleaning. Ad-hoc randomization is achieved by using *col1+col1.col2+row1+row1.row2* as the randomization model. This formula indicates that randomization is defined by:

OUTPUT	**COMMENT**
Class 0, pseudofactors *col1*	Randomization of macro-columns
]0)= , associated pseudofactors:	No nesting pseudofactors
fonction fi : 1 0	permutation : $0 \mapsto 1$, $1 \mapsto 0$
Class 1, pseudofactors *row1*	Randomization of macro-rows
]1)= , associated pseudofactors:	No nesting pseudofactors
fonction fi : 1 0	permutation : $0 \mapsto 1$, $1 \mapsto 0$
Class 2, pseudofactors *col2*	randomization of columns
]2)=0 , associated pseudofactors: *col1*	in each macro-column
function fi : 0 0 1 1 1 0	permutation macro-column 0 : $0 \mapsto 0$, $1 \mapsto 1$ macro-column 1 : $0 \mapsto 1$, $1 \mapsto 0$
Class 3, pseudofactors *row2*	randomization of rows
]3)=1 , associated pseudofactors: *row1*	in each macro-row
function fi : 0 0 1 1 0 1	permutation macro-row 0 : $0 \mapsto 0$, $1 \mapsto 1$ macro-row 1 : $0 \mapsto 0$, $1 \mapsto 1$

Table 12: Randomization of the ROBPL1R4.PS design (intermediate outputs)

- a random permutation of the macro-columns (*col1*),
- a random permutation of the macro-rows (*row1*),
- for each of the macro-columns, a random permutation of the columns of which it consists (*col2* for *col1* fixed),
- for each of the macro-rows, a random permutation of the two rows of which it consists (*row2* for *row1* fixed).

Random column permutations are conducted independently in the two macro-columns. Likewise, row permutations are conducted independently in the two macro-rows. An illustration of the possible permutations is given in Table 12, and R code to get a randomized design is given below.

```
> robpl1r4.rand <- planor.design(robpl1r4.key,
+                                randomize=~col1/col2 + row1/row2)
> print(robpl1r4.rand@design)
```

	row1	row2	col1	col2	nsoil	qsoil	cbact	Tact	conc	brush	rough	nat
1	1	1	1	1	Saint-Paulin	10mg	6%	15mn	3%	strong	0.75	1
5	1	2	1	1	Saint-Paulin	10mg	3%	15mn	1%	weak	0.75	2
9	2	1	1	1	Saint-Paulin	10mg	3%	30mn	3%	weak	0.25	1
13	2	2	1	1	Saint-Paulin	10mg	6%	30mn	1%	strong	0.25	2
2	1	1	1	2	Saint-Paulin	100mg	3%	15mn	3%	strong	0.25	2
6	1	2	1	2	Saint-Paulin	100mg	6%	15mn	1%	weak	0.25	1
10	2	1	1	2	Saint-Paulin	100mg	6%	30mn	3%	weak	0.75	2
14	2	2	1	2	Saint-Paulin	100mg	3%	30mn	1%	strong	0.75	1
3	1	1	2	1	curd	100mg	3%	15mn	3%	weak	0.75	1
7	1	2	2	1	curd	100mg	6%	15mn	1%	strong	0.75	2
11	2	1	2	1	curd	100mg	6%	30mn	3%	strong	0.25	1
15	2	2	2	1	curd	100mg	3%	30mn	1%	weak	0.25	2
4	1	1	2	2	curd	10mg	6%	15mn	3%	weak	0.25	2
8	1	2	2	2	curd	10mg	3%	15mn	1%	strong	0.25	1
12	2	1	2	2	curd	10mg	3%	30mn	3%	strong	0.75	2
16	2	2	2	2	curd	10mg	6%	30mn	1%	weak	0.75	1

What purpose does this randomization serve? Past experiments seem to show that the use of controls makes the effect *position of the test specimen on the plate* negligible. There is, however, no certainty that this is always the case and prudence demands that the systematic use of the very regular design obtained prior to randomization be avoided.

In principle, the theoretical study of this type of randomization leads to selecting a statistical analysis model which includes all the block effects satisfying the hierarchies, that is to say, that never dissociate column *col2* from macro-column *col1* and, similarly, row *row2* from macro-row *row1*. The terms of this model:

col1; row1; row1.col1; col2.col1; col2.col1.row1; row2.row1; row2.row1.col1; row2.row1.col2.col1

are the *ancestral terms*. Randomization makes these block effects random and, to be entirely accurate, the analysis must take this into account to test each treatment effect in the appropriate "*strata*".

In this example, each block effect is confounded with one or several treatment effects, as indicated by running the `alias` function on `robpl1r4.key` with model

`col1.col2.row1.row2 + p.p`,

where p denotes the addition of the main effects of the 8 treatment factors. It is therefore out of question to perform the analysis by the decomposition into strata mentioned above. For the purpose of the analysis, we have to rely on the assumption that there is no effect of position on the plate. Neither are there, in fact, any residual degrees of freedom that

would enable estimation of error variance. Therefore, the analysis relies on the assumption that some of the linear combinations of aliased treatment effects given by the code below are equal to zero, and the others must be detected by a procedure such as those described in [18] or [20].

```
> alias(robpl1r4.key,
+       model=~row1*row2*col1*col2 +
+              (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2)


********** Prime  2  design **********

--- Solution  1  for prime  2  ---

UNALIASED TREATMENT EFFECTS
nil

ALIASED TREATMENT EFFECTS
nil

TREATMENT AND BLOCK EFFECTS CONFOUNDED WITH BLOCK EFFECTS
row1 = Tact
row2 = nsoil:brush = qsoil:cbact = Tact:conc = rough:nat
col1 = nsoil
col2 = qsoil
cbact = row2:col2
conc = row1:row2
brush = row2:col1
rough = row1:col1:col2
nat = row1:row2:col1:col2
row1:col1 = nsoil:Tact = qsoil:rough = cbact:nat = conc:brush
row1:col2 = nsoil:rough = qsoil:Tact = cbact:conc = brush:nat
col1:col2 = nsoil:qsoil = cbact:brush = Tact:rough = conc:nat
nsoil:cbact = qsoil:brush = Tact:nat = conc:rough = row2:col1:col2
nsoil:conc = qsoil:nat = cbact:rough = Tact:brush = row1:row2:col1
nsoil:nat = qsoil:conc = cbact:Tact = brush:rough = row1:row2:col2

UNALIASED BLOCK EFFECTS
nil


--- Synthesis on the aliased treatment effects for prime  2  ---

     unaliased trt.aliased blc.aliased
[1,]         0           0          36
```

3.2.2 Resolution 5 design for 5 factors

A design of resolution 5 for 5 factors can be searched for by the code just below. The factor intensity of brushing is not studied in this design, therefore, it is unnecessary to decompose the columns into two macro-columns as in the previous example.

```
> robpl1r5.fac <- planor.factors(factors=list(
+                                  row1=1:2, row2=1:2, col=1:4,
+                                  nsoil=c("curd","Saint-Paulin"),
+                                  #qsoil=c("10mg","100mg"),
+                                  cbact=c("3%","6%"),
+                                  Tact=c("15mn","30mn"),
+                                  conc=c("1%","3%"),
+                                  #brush=c("strong","weak"),
+                                  rough=c(0.25,0.75)),
+                                  #nat=1:2),
+                                block=~row1+row2+col,
+                                hierarchy=list(~nsoil/(col*row2),
+                                  ~cbact/(col*row2),
+                                  ~Tact/(row1*row2),
+                                  ~conc/(row1*row2)))
>                                  #~brush/(row1*row2*col1)))
>
> robpl1r5.mod <- planor.model( listofmodels=list(
+  c(~(nsoil+cbact+Tact+conc+rough)^2, ~(nsoil+cbact+Tact+conc+rough)^2),
+  c(~row2 + nsoil+cbact+Tact+conc+rough, ~nsoil+cbact+Tact+conc+rough),
+  c(~col + rough, ~rough) ) )
> robpl1r5.key <- planor.designkey(factors=robpl1r5.fac,
+                                  model=robpl1r5.mod,
+                                  nunits=16,
+                                  base=~row1+row2+col)

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 5 to 9
      first visit to column 5
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
The search is closed: max.sol =  1 solution(s) found

>
```

The first model indicates that all the main effects and two-factor interactions of the treatments factors are sought to be estimated.

It is impossible to estimate the 16 parameters in a model that also includes a block effect such as soiling effect *row2*. In contrast, as in the previous example, it is possible to ensure that all the main effects are estimable in a model including soiling effect (*row2*) and that roughness is also estimable within columns, that is to say, in the presence of a term *col* in the model. The latter two constraints relate to the second and third pairs *model—part to be estimated* in the `listofmodels` argument of `planor.model`. The substitution of *col.row2* for *col* in pair 3 would produce the following diagnosis, which proves that it is not possible to impose the same constraint on roughness as in the previous example:

```
> robpl1r5.modbis <- planor.model( listofmodels=list(
+  c(~(nsoil+cbact+Tact+conc+rough)^2, ~(nsoil+cbact+Tact+conc+rough)^2),
+  c(~row2 + nsoil+cbact+Tact+conc+rough, ~nsoil+cbact+Tact+conc+rough),
+  c(~col*row2 + rough, ~rough) ) )
> robpl1r5bis.key <- planor.designkey(factors=robpl1r5.fac,
+                                     model=robpl1r5.modbis,
+                                     nunits=16,
+                                     base=~row1+row2+col)

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 5 to 9
      first visit to column 5
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
The search is closed:  0 solutions found


>
```

The solution obtained is printed below, as well as the study of aliases with model *col.row1.row2+p.p*. There are no degrees of freedom to estimate residual variance, therefore statistical analysis must be based on a method of detecting influencing effects similar to that in the previous example.

```
> #summary(robpl1r5.key)
> alias(robpl1r5.key,
+       model=~col*row1*row2 + (nsoil+cbact+Tact+conc+rough)^2)


********** Prime  2  design **********

--- Solution  1  for prime  2  ---

UNALIASED TREATMENT EFFECTS
```

```
nil

ALIASED TREATMENT EFFECTS
nil

TREATMENT AND BLOCK EFFECTS CONFOUNDED WITH BLOCK EFFECTS
col_1 = nsoil
col_2 = cbact
col_1:col_2 = nsoil:cbact
row1 = Tact
row2 = Tact:conc
conc = row1:row2
rough = row2:col_1:col_2
row1:col_1 = nsoil:Tact
row1:col_2 = cbact:Tact
row1:col_1:col_2 = conc:rough
row2:col_1 = cbact:rough
row2:col_2 = nsoil:rough
nsoil:conc = row1:row2:col_1
cbact:conc = row1:row2:col_2
Tact:rough = row1:row2:col_1:col_2

UNALIASED BLOCK EFFECTS
nil


--- Synthesis on the aliased treatment effects for prime  2  ---

     unaliased trt.aliased blc.aliased
[1,]         0           0          15
```

3.3 Designs for two plates

We know that with 32 units, it is possible to study up to six 2-level factors in resolution 5 and up to 16 in resolution 4.

If one of the studied factors is qualitative with 4 levels, as the others still have 2 levels, the situation is quite different. For a resolution of 5, there may be up to four 2-level factors in addition to the 4-level factor. For a resolution of 4, up to seven 2-level factors can be introduced.

A maximum of 7 is easily obtained by the search defined just below. The result of this search reveals the impossibility of going beyond the 7th factor.

```
> size32r4.fac <- planor.factors(factors=LETTERS[1:12],
+                                nlevels=c(4, rep(2,11)))
> size32r4.mod <- planor.model(model="(A+B+C+D+E+F+G+H+I+J+K+L)^2,
```

```
+                                estimate=~A+B+C+D+E+F+G+H+I+J+K+L)
> size32r4.key <- planor.designkey(factors=size32r4.fac,
+                                  model=size32r4.mod,
+                                  nunits=32,
+                                  base=~A+B+C+D)


size32r4.key <- planor.designkey(factors=size32r4.fac,
+                                  model=size32r4.mod,
+                                  nunits=32,
+                                  base=~A+B+C+D)
Determination of ineligible factorial terms
Determination of ineligible pseudofactorial terms
Independent searches for prime(s) :  2
Key-matrix search for prime p = 2
There are  5  predefined columns
First visit to column  6
First visit to column  7
First visit to column  8
First visit to column  9
First visit to column  10
  C-c C-c
```

This search introduces, as basic factors, factor A with 4 levels and three other factors with 2 levels. It may be proved simply that this choice is not restrictive by using the following result: if factors cannot be chosen as basic factors, a defining relation exists between them. Thus if neither of the two sets $\{A, B, C, D\}$, $\{A, B, C, E\}$ can be used as a basic set, two relations of the form $A_1^\alpha A_2^\beta BCD = 1$, $A_1^\gamma A_2^\delta BCE = 1$ may be formed. Nevertheless, multiplying these relations gives rise to a relation between the three factors A, D, E which cannot exist in resolution 4.

In resolution 5, the fact that it is impossible to go beyond four 2-level factors results from an elementary calculation of the number of *degrees of freedom*. With five 2-level factors, the model includes, in addition to the general mean, $3 + 5$ parameters associated with the main effects and $3 \times 5 + 5 \times 4/2 = 25$ parameters associated with the interactions. This amounts to a total of 34 parameters, which exceeds the number of experimental units.

If the single 4-level factor is quantitative, the situation is different though. In this case, for the purpose of the analysis we refer to [18], [7] and to paragraph 4 relating to designs for combinations of 2- and 4-level factors.

In the case of the robot, the following examples illustrate the above cases.

3.3.1 Resolution 5 fraction $2^6/2$

The input parameters for such a design are given below, as well as the solution in the form of a key matrix.

```
> robpl2r5.fac <- planor.factors(factors=list(
+                                  plate=1:2,
+                                  row1=1:2, row2=1:2, col=1:4,
+                                  nsoil=c("curd","Saint-Paulin"),
```

```
+                                    qsoil=c("10mg","100mg"),
+                                    cbact=c("3%","6%"),
+                                    Tact=c("15mn","30mn"),
+                                    conc=c("1%","3%"),
+                                    #brush=c("strong","weak"),
+                                    rough=c(0.25,0.75)),
+                                    #nat=1:2),
+                                  block=~plate+row1+row2+col,
+                                  hierarchy=list(~nsoil/(col*row2),
+                                    ~cbact/(col*row2),
+                                    ~Tact/(row1*row2),
+                                    ~conc/(row1*row2)))
>                                   #~brush/(row1*row2*col1)))
>
> robpl2r5.mod <- planor.model( listofmodels=list(
+   c( ~plate + (nsoil+qsoil+cbact+Tact+conc+rough)^2,
+      ~plate + (nsoil+qsoil+cbact+Tact+conc+rough)^2),
+   c( ~plate*row2 + nsoil+qsoil+cbact+Tact+conc+rough,
+      ~             nsoil+qsoil+cbact+Tact+conc+rough),
+   c( ~plate*col*row2 + rough,
+      ~                 rough) ) )
> robpl2r5.key <- planor.designkey(factors=robpl2r5.fac,
+                                  model=robpl2r5.mod,
+                                  nunits=32,
+                                  base=~plate+row1+row2+col)
> summary(robpl2r5.key)
> alias(robpl2r5.key,
+       model=~plate + (nsoil+qsoil+cbact+Tact+conc+rough)^2)
```

Hierarchical constraints are similar in nature to those in the designs for one plate. Estimation of the plate effect is required (first pair *model — part to be estimated*). It is also ensured that all the main effects are estimable, including in the presence of a soiling effect (second pair) and that roughness varies within columns when both specimens are soiled simultaneously (third pair).

The design obtained permits estimation of the constant, the plate effect, the 6 main effects and 15 two-factor interactions, under the assumption that there are no interactions of three factors or more. For the estimation of residual variance, it leaves $9 = 32 - (1 + 1 + 6 + 15)$ degrees of freedom, which correspond to the unaliased block effects different from *plate* obtained through the study of aliases for model *pl.col.row1.row2 + p.p.*

3.3.2 A $2^8/8$ fraction of resolution 4

In this example, twenty different solutions are required and the `randomsearch` argument of `planor.designkey` is set to `TRUE`. In this case, the program orders randomly the set of possible choices for each column of the key matrix to be defined.

```
> robpl2f8r4.fac <- planor.factors(factors=list(
+                                  plate=1:2,
+                                  row1=1:2, row2=1:2,
+                                  col1=1:2, col2=1:2,
+                                  nsoil=c("curd","Saint-Paulin"),
+                                  qsoil=c("10mg","100mg"),
+                                  cbact=c("3%","6%"),
+                                  Tact=c("15mn","30mn"),
+                                  conc=c("1%","3%"),
+                                  brush=c("strong","weak"),
+                                  rough=c(0.25,0.75),
+                                  nat=1:2),
+                                  block=~plate+row1+row2+col1+col2,
+                                  hierarchy=list(
+                                    ~nsoil/(plate*col1*col2*row2),
+                                    ~cbact/(plate*col1*col2*row2),
+                                    ~Tact/(plate*row1*row2),
+                                    ~conc/(plate*row1*row2),
+                                    ~brush/(plate*row1*row2*col1)))
> robpl2f8r4.mod <- planor.model( listofmodels=list(
+   c( ~plate + (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2,
+      ~plate +  nsoil+qsoil+cbact+Tact+conc+brush+rough+nat),
+   c( ~plate*row2 + nsoil+qsoil+cbact+Tact+conc+brush+rough+nat,
+      ~             nsoil+qsoil+cbact+Tact+conc+brush+rough+nat),
+   c( ~plate*col1*col2*row2 + rough+nat,
+      ~                       rough+nat) ) )
> robpl2f8r4.key <- planor.designkey(factors=robpl2f8r4.fac,
+                                    model=robpl2f8r4.mod,
+                                    nunits=32,
+                                    base=~plate+row1+row2+col1+col2,
+                                    max.sol=20,
+                                    randomsearch=TRUE)
> summary(robpl2f8r4.key)
> alias(robpl2f8r4.key,
+       model=~plate*row1*row2*col1*col2 +
+         (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2)
```

In order to compare solutions, the study of aliases is performed with model

```
~plate + (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2.
```

Table 13 provides the outputs for three possible solutions 3, 7, 9. These outputs in fact illustrate the three types of results obtained through the study of aliases for the twenty solutions.

The first kind of solution represented by solution 3 provides 7 sets of aliased treatment effects, one of which consists of 3 two-factor interactions and the other 6 consisting

Set of aliased treatment effects		
Solution 3	Solution 7	Solution 9
rough nat ; n-soil c-bact ; q-soil conc ; conc rough ; q-soil nat ; conc nat ; q-soil rough ; c-bact conc ; n-soil q-soil ; n-soil conc ; q-soil c-bact ; n-soil nat ; c-bact rough ; n-soil rough ; c-bact nat ;	q-soil rough ; T-act nat ; conc brush ; n-soil c-bact ; T-act rough ; q-soil nat ; q-soil T-act ; rough nat ; conc rough ; q-soil brush ; q-soil conc ; brush rough ; T-act conc ; brush nat ; conc nat ; T-act brush ; q-soil c-bact ; n-soil nat ; c-bact rough ; n-soil T-act ; n-soil q-soil ; c-bact nat ; n-soil rough ; c-bact T-act ;	brush nat ; n-soil T-act ; q-soil c-bact ; q-soil nat ; c-bact brush ; T-act conc ; q-soil brush ; c-bact nat ; n-soil conc ; q-soil T-act ; n-soil c-bact ; conc nat ; n-soil q-soil ; c-bact T-act ; conc brush ; n-soil brush ; T-act nat ; q-soil conc ; n-soil nat ; T-act brush ; c-bact conc ;
Treatment effects unaliased with other treatment effects		
Solution 3	Solution 7	Solution 9
n-soil ; conc brush ; T-act nat ; >c-bact ; q-soil brush ; T-act rough ; >q-soil T-act ; brush rough ; conc ; >brush nat ; T-act brush ; T-act; >q-soil ; brush ; n-soil brush ; >T-act conc ; c-bact brush ; nat ; >n-soil T-act ; rough ; c-bact T-act ;	c-bact ; n-soil ; rough ; q-soil ; >T-act ; nat ; c-bact brush ; >conc ; brush ; c-bact conc ; >n-soil brush ; n-soil conc ;	n-soil ; T-act ; T-act rough ; >n-soil rough ; rough ; conc rough ; conc ; >c-bact ; q-soil ; q-soil rough ; >c-bact rough ; rough nat ; brush rough ; >brush ; nat ;
Block effects unaliased with a treatment effect		
Solution 3	Solution 7	Solution 9
pl ; row1 ; pl row1 row2 col1 ;	row2 ; pl row2 col2 ; col2 ; >row2 col1 col2 ; pl ; >pl row2 col1 ; pl row1 row2 col2 ; >pl row1 col1 col2 ;	row2 ; pl ; pl row1 ; >pl row1 col2 ; pl row2 col2 ; >row2 col1 col2 ; row1 row2 col1 ; >pl col1 col2 ; pl row1 col1 ;

Table 13: Aliasing in three typical resolution 4 solutions for a $2^8/8$

The sign > indicates a continuation of the same row.

of only two interactions. It therefore leads to 15 aliased interactions and $13 = C_8^2 - 15$ unaliased interactions. This type of design thus enables (orthogonal) estimation of 8 main effects, 13 isolated interactions and 7 linear combinations of interactions. The presence of *plate* in the first estimate formula also makes it possible to ensure that the plate effect is estimable. Following estimation of these $29 = 8 + 13 + 7 + 1$ effects and of the constant, only $2 = 32 - 30$ degrees of freedom remain to estimate residual variance – degrees of freedom which correspond to the two unaliased block effects different from *plate* obtained by the analysis of aliases (bottom of table 13). This low number of degrees of freedom makes it necessary either to combine this estimation with a previously obtained estimation or to use a procedure which detects influencing factors such as that mentioned earlier.

The second kind of solution represented by solution 7 provides 11 sets of aliased treatment effects including two sets of 3 interactions and 9 sets of two interactions, that is to say, 24 aliased and 4 unaliased interactions. It thus allows to estimate the constant, the plate effect, the 8 main effects, 4 isolated interactions, 11 linear combinations of interactions and leaves $7 = 32 - (1 + 1 + 8 + 4 + 11)$ degrees of freedom to estimate the residual variance. These 7 degrees of freedom correspond to the unaliased block effects different from *plate* obtained by the analysis of aliases (bottom of table 13).

Finally, the third kind of solution represented by solution 9 enables the estimation of 7 isolated interactions, 7 linear combinations of three interactions and leaves 8 degrees of freedom to estimate residual variance.

The first type of design makes it possible to estimate more interactions and is therefore preferable to the other two types of design, except probably in cases in which many effects are expected to be significant and in which the experiment is intended to provide a correct estimator of residual variance.

The specific choice of design within the first type of design is unimportant if the same importance is attached *a priori* to all two-factor interactions. For the purpose of the example, we selected solution 3, but other equivalent solutions exist.

3.3.3 Fraction $4 \times 2^4/2$ of resolution 5

The model now contains the constant, the *plate* effect, the main effect of the 4-level qualitative factor *n-soil*, the 4 main effects of 2-level factors and their 6 interactions, the 12 interaction parameters between factor *n-soil* and each of the other 4 factors. It thus contains a total of $27 = 1 + 1 + 3 + 4 + 6 + 12$ parameters and leaves $5 = 32 - 27$ degrees of freedom to estimate the error variance. These degrees of freedom correspond to the unaliased block effects different from *plate* appearing at the bottom of table 14. They were obtained by the analysis of aliases. A possible solution is presented in Table 14.

```
> robpl2f4r5.fac <- planor.factors(factors=list(
+                                  plate=1:2,
+                                  row1=1:2, row2=1:2, col=1:4,
+                                  nsoil=1:4,
+                                  cbact=c("3%","6%"),
```

```
+                                    Tact=c("15mn","30mn"),
+                                    conc=c("1%","3%"),
+                                    rough=c(0.25,0.75)),
+                                  block=~plate+row1+row2+col,
+                                  hierarchy=list(
+                                    ~nsoil/(plate*col*row2),
+                                    ~cbact/(plate*col*row2),
+                                    ~Tact/(plate*row1*row2),
+                                    ~conc/(plate*row1*row2)))
> robpl2f4r5.mod <- planor.model( listofmodels=list(
+   c( ~plate + (nsoil+cbact+Tact+conc+rough)^2,
+                         ~plate + (nsoil+cbact+Tact+conc+rough)^2),
+   c( ~plate*row2 + cbact+Tact+conc+rough,
+                         ~cbact+Tact+conc+rough) ) )
> robpl2f4r5.key <- planor.designkey(factors=robpl2f4r5.fac,
+                                    model=robpl2f4r5.mod,
+                                    nunits=32,
+                                    base=~plate+row1+row2+col)
> summary(robpl2f4r5.key)
> alias(robpl2f4r5.key, model=~plate*row1*row2*col
+                               + (nsoil+cbact+Tact+conc+rough)^2)
```

Design definition

key matrix and defining relations

	n-soil_1	n-soil_2	c-bact	T-act	conc	rough
pl	0	0	1	1	0	0
row1	0	0	0	1	1	0
row2	0	1	0	0	1	1
col_1	1	1	1	0	0	0
col_2	0	0	1	0	0	1

n-soil_1	=	col_1
n-soil_2	=	row2 col_1
c-bact	=	pl col_1 col_2
T-act	=	pl row1
conc	=	row1 row2
rough	=	row2 col_2

list of unaliased block effects

pl ; pl col_1 ; pl row1 col_2 ; pl row1 col_1 col_2 ; row1 row2 col_2 ; row1 row2 col_1 col_2 ;

Table 14: Design ROP2F4R5 for 2 plates, a $4 \times 2^4/2$ of resolution 5

3.3.4 Design $4 \times 2^7/16$ of resolution 4

Before examining the adaptation of such a design to the robot, a search may be performed to find out which types of design of this form may be obtained. The analysis of aliases for the 20 solutions obtained by the search defined below reveals a single alias structure and suggests that, allowing for a possible permutation of the 2-level factors and also of the three pseudofactors A_1, A_2, A_1A_2, there is only one solution. Table 15 specifies the confounding for one of the solutions. It shows 18 unaliased effects and 13 sets of aliased interactions. Such a set-up does not leave any residual degree of freedom and does not enable estimation of a potential additional block effect.

```
> n32f4and7f2r4.fac <- planor.factors(factors=LETTERS[1:8],
+                                     nlevels=c(4, rep(2,7)))
> n32f4and7f2r4.mod <- planor.model( model=~(A+B+C+D+E+F+G+H)^2,
+                                   estimate=~A+B+C+D+E+F+G+H )
> n32f4and7f2r4.key <- planor.designkey(factors=n32f4and7f2r4.fac,
+                                       model=n32f4and7f2r4.mod,
+                                       nunits=32,
+                                       base=~A+B+C+D,
+                                       max.sol=20,
+                                       randomsearch=TRUE)
> summary(n32f4and7f2r4.key)
> alias(n32f4and7f2r4.key, model=~(A+B+C+D+E+F+G+H)^2)
```

Detailed list of sets of aliased effects in the model
A_1 B ; E F ; A_1 A_2 B ; D H ; C G ; A_1 C ; E H ; A_1 A_2 C ; D F ; B G ; B C ; F H ; A_1 A_2 G ; D E ; A_1 G ; A_1 D ; E G ; A_1 A_2 D ; B H ; C F ; B D ; F G ; A_1 A_2 H ; A_1 H ; C E ; C D ; G H ; A_1 A_2 F ; A_1 F ; B E ; C H ; B F ; A_1 E ; D G ;
list of unaliased effects
; A_1 ; A_2 ; A_1 A_2 ; B ; A_2 B ; C ; A_2 C ; A_2 G ; G ; >D ; A_2 D ; A_2 H ; H ; A_2 F ; F ; A_2 E ; A_1 A_2 E ; E;

Table 15: Alias structure in a $4 \times 2^7/16$ of resolution 4

The R code below generates this type of design for the robot. Since it is impossible to estimate *plate*, it is precluded from being included in the part to be estimated. One of the possible solutions is presented in Table 16.

```
> rop2f4r4.fac <- planor.factors(factors=list(
+                                plate=1:2,
+                                row1=1:2, row2=1:2,
+                                col1=1:2, col2=1:2,
+                                nsoil=1:4,
+                                qsoil=c("10mg","100mg"),
+                                cbact=c("3%","6%"),
+                                Tact=c("15mn","30mn"),
+                                conc=c("1%","3%"),
```

```
+                                    brush=c("strong","weak"),
+                                    rough=c(0.25,0.75),
+                                    nat=1:2),
+                                    block=~plate+row1+row2+col1+col2,
+                                    hierarchy=list(
+                                      ~nsoil/(plate*col1*col2*row2),
+                                      ~cbact/(plate*col1*col2*row2),
+                                      ~Tact/(plate*row1*row2),
+                                      ~conc/(plate*row1*row2),
+                                      ~brush/(plate*row1*row2*col1)))
> rop2f4r4.mod <- planor.model(
+   model= ~plate + (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2,
+   estimate= ~      nsoil+qsoil+cbact+Tact+conc+brush+rough+nat )
> rop2f4r4.key <- planor.designkey(factors=rop2f4r4.fac,
+                                    model=rop2f4r4.mod,
+                                    nunits=32,
+                                    base=~plate+row1+row2+col1+col2)
> summary(rop2f4r4.key)
> alias(rop2f4r4.key,
+   model=~plate*row1*row2*col1*col2
+         + (nsoil+qsoil+cbact+Tact+conc+brush+rough+nat)^2)
```

Design definition

key matrix

	$n\text{-}soil_1$	$n\text{-}soil_2$	q-soil	c-bact	T-act	conc	brush	rough	nat
pl	1	0	0	0	0	1	0	1	1
row1	0	0	1	0	1	1	1	0	0
row2	1	0	0	1	1	1	1	0	1
col1	1	1	1	1	0	0	1	0	0
col2	1	1	0	1	0	0	0	1	1

defining relations

$n\text{-}soil_1$	=	pl row2 col1 col2
$n\text{-}soil_2$	=	col1 col2
q-soil	=	row1 col1
c-bact	=	row2 col1 col2
T-act	=	row1 row2
conc	=	pl row1 row2
brush	=	row1 row2 col1
rough	=	pl col2
nat	=	pl row2 col2

list of unaliased block effects

pl ; pl col_1 ; pl row1 col_2 ; pl row1 col_1 col_2 ; row1 row2 col_2 ; row1 row2 col_1 col_2 ;

Table 16: Design ROP2F4R4 for 2 plates, a $4 \times 2^8/32$ of resolution 4

4 Design for a combination of 2 and 4-level factors

4.1 Qualitative factors

4.1.1 Example

In a study on the effectiveness of cleaning and disinfection of surfaces, 16 factors are initially identified. They include four important qualitative factors, such as the type of disinfection product, for which four modalities are required.

To compare different treatments, experiments are conducted on samples of surfaces treated in blocks of size 8. The 2 levels 4° et 20° of the factor *cleaning temperature* T are necessarily constant within each block.

In order to determine the most influential factors, 64 treatments corresponding to 8 blocks may be tested in a first step. With a view to minimizing the number of trials, it is decided to study only two levels for factors other than the four already mentioned. As regards the latter, four levels are selected rather than three. This enables a broader range of treatments to be covered, and substantially increases possibilities of creating design fractions.

Initially, the blocks are ignored. The maximum number of 2-level factors which may be added to the four 4-level factors is sought. The resolution is limited to 3 or 4, because resolution 5 is unattainable with four 4-level factors.

```
> n64w4f4f2r4.key <- planor.designkey(
+   factors=LETTERS[1:17],
+   nlevels=c(rep(4,4), rep(2,13)),
+   model="(A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q)^2,
+   estimate="A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q,
+   nunits=64, base="A+B+C)
> n64w3f4f2r4.key <- planor.designkey(
+   factors=LETTERS[1:17],
+   nlevels=c(rep(4,3), rep(2,14)),
+   model="(A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q)^2,
+   estimate="A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q,
+   nunits=64, base="A+B+C)
> n64w2f4f2r4.key <- planor.designkey(
+   factors=LETTERS[1:17],
+   nlevels=c(rep(4,2), rep(2,15)),
+   model="(A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q)^2,
+   estimate="A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q,
+   nunits=64, base="A+B+C+D)
> n64w1f4f2r4.key <- planor.designkey(
+   factors=LETTERS[1:17],
+   nlevels=c(rep(4,1), rep(2,16)),
+   model="(A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q)^2,
```

```
+    estimate=~A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q,
+    nunits=64, base=~A+B+C+D+E)
>
```

There are necessarily defining relations linking the four 4-level factors because a design of size 64 can only contain a fraction of the $256 = 4^4$ combinations of levels of these four factors.

Within seconds, the search for a design of resolution 4 finds relations which make it possible to define 4 two-level factors E, F, G, H in addition to the four 4-level factors A, B, C, D. The search for a 5th factor I has still not been completed after 5 mn and is abandoned. This is indicated by the fact that the backtrack algorithm makes its first visit to columns 7 (pseudofactor D_1) to 13 (factor I) rapidly, but cannot find a correct column 13.

Two possibilities arise for further study : reducing the number of 4-level factors or limiting the resolution to 3.

4.1.2 Resolution 4 designs

A design of resolution 4 is a design in which all the main effects are estimable in a model which includes, alongside the constant and the main effects, all the two-factor interactions. Through suitable reparametrization, such as for instance that described in [22], the following result is easily demonstrated, in which X is the linear model matrix.

Proposition 4.1 *In a design of resolution* 4*, the columns X associated with the constant, the main effects and the interactions between a fixed factor and each of the other factors are independent.*

This result was demonstrated by Margolin [23] by using the reparametrization associated with orthogonal polynomials. It provides a lower bound to the number N of units enabling the construction of a design of resolution 4 with predetermined factors: N must be greater than the number of independent columns provided by the proposal.

Thus in the case at hand, if there are n_4 four-level factors, n_2 two-level factors, the number of columns associated with the main effects is $n_2 + 3n_4$. If the determined factor has 4 levels, the number of columns associated with the interactions between this factor and each of the $n_2 + n_4 - 1$ others is $3(n_2 + 3(n_4 - 1))$. We should thus have

$$N \geq 1 + n_2 + 3n_4 + 3(n_2 + 3(n_4 - 1)) \; .$$

When N and n_4 are fixed, this formula provides the following upper bound for the number n_2 of two-level factors introducible in resolution 4 :

$$n_2 \leq \frac{N}{4} - 3n_4 + 2 \; . \tag{6}$$

In the cases $\{n_4 = 1,\ N = 16k\}$ and $\{n_4 = 2,\ N = 32k\}$, it is known how to construct designs of resolution 4 with this maximum number of n_2 factors with 2 levels [23], [1].

These constructions, based on the Hadamard matrices, do not guarantee regularity and, consequently, simplicity of confounding between two-factor interactions, but they enable the development of designs with a number of units which is not a power of 2 (for instance $n_4 = 1$ and $N = 48$, or $n_4 = 2$, $N = 96$).

In contrast, for $n_4 = 3$, there is no known comparable general method and only one algorithm such as that of `PLANOR` enables the development of orthogonal designs with a substantial number of two-level factors.

The results obtained by `PLANOR` with $N = 64$ and n_4 varying between 1 and 4 and $N = 32$ are summarized in table 17. This table specifies the basic factors used in each search.

The choices made for these basic factors are not restrictive. This is clear when the basic factors are the four-level factors A, B, C because in resolution 4 there cannot be any defining relation linking these three factors. When there are only two 4-level factors A and B, it is certainly possible to add C to them. Resolution 4 precludes any defining relation between these three factors and as a result the 32 combinations of their levels appear. If another 2-level factor could not be added to them, all the other factors would subsequently be defined from A, B, C, and the design would consist of two replications of the same 32 treatments, which must be avoided. Similar reasoning applies in cases in which there is only one 4-level factor.

---

nb. of 4-lev. factors	maximum nb. of 2-lev. factors	basic factors	added factors after a 5 mn search
4	4	$A_1\ A_2\ B_1\ B_2\ C_1\ C_2$	$D_1\ D_2\ E\ F\ G\ H\quad J\ K\ L\ M\ N\ O\ P\ Q$
3	7	$A_1\ A_2\ B_1\ B_2\ C_1\ C_2$	$D\ E\ F\ G\ H\ I\ J\quad L\ M\ N\ O\ P\ Q$
2	12	$A_1\ A_2\ B_1\ B_2\ C\ D$	$E\ F\ G\ H\ I\ J\ K\ L\ M\ N\quad P\ Q$
1	15	$A_1\ A_2\ B\ C\ D\ E$	$F\ G\ H\ I\ J\ K\ L\ M\ N\ O\ P$

Table 17: Resolution 4 designs for a mixture of 4 and 2 levels qualitative factors

The asterisk indicates the factor which the program was attempting to define after searching for 5 mn. In general, this factor is obtained in a fraction of a minute. The number appearing in the second column is the number of 2-level factors on the left-hand side of $\star$, that is to say the maximum number which may be defined within a reasonable time frame

The maximum number of 2-level factors given in this table 17 is to be compared with the upper bounds from (6) which are specified for n_4 included between 1 and 6 in Table 18.

n_4	1	2	3	4	5	6
n_2 max	15	12	9	6	3	0

Table 18: Margolin upper bound for n_2 in a size 64 resolution 4 fraction of a $4^{n_4}2^{n_2}$

The maximum number of 2-level factors obtained within a reasonable time frame is slightly below the maximum provided in Table 18 when $n_4 = 4$ or $n_4 = 3$. It is equal to this maximum in cases $n_4 = 2$ and $n_4 = 1$.

Resolution 4 provides correct estimates of the main effects, included in the presence of interactions. It therefore makes it possible to draw generally quite reliable conclusions. However, table 17 shows that obtaining this resolution requires substantially restricting the targets initially set with regard to the number of factors studied.

4.1.3 Resolution 3 designs giving a resolution 4 by foldover

If the existence of a small number of really active factors is suspected without further specific indications, a good strategy can be to increase the number of factors studied by reducing the resolution to 3.

The maximum number of 2-level factors which may be added to the 4-level factors in resolution 3 is easy to find. The requirement is not to have a defining relation that links only two factors. For instance, if A, B, C, D are the four-level factors, E, F, ... the two-level factors, there should be no relations such as $B_1 = A_1A_2$, $E = A_1A_2$, $E = F$. In other words, the elements A_1, A_2, A_1A_2, B_1, B_2, B_1B_2, C_1, C_2, C_1C_2, D_1, D_2, D_1D_2, E, F, ... must all be different. With 64 units, there are at most 63 different non-zero elements formed from the basic factors. Therfore, once the 4-level factors are determined, there remain $51 = 63 - 12$ elements which may be used to define the 2-level factors. More generally, when s_4 is the number of 4-level factors, the maximum number of 2-level factors which may be added is $s_2 = 63 - 3 \times s_4$.

Moreover, the theory gives the maximum number s_4 of 4-level factors which may be introduced beforehand: $21 = 63/3$ (number of vectorial subspaces of dimension 1 in the vectorial space F_4^3 of dimension 3 on the Galois field with 4 elements).

The probability of detecting the active factors increases with the number of factors studied, but the decrease in the resolution considerably weakens the conclusions which may be drawn. Such a design of resolution 3 is therefore generally only envisaged as an intermediate stage which enables rapid detection of the most important factors. It may be followed by another design restricted to the active factors detected, in order to prove or disprove the results found and to study the interactions. Another possible continuation, useful when the number of significant effects prevents interpretation owing to possible confounding with the interactions, is a *design foldover* which in total leads to a design of resolution 4.

4.1.3.1 Complete foldover This possibility of *duplication* by foldover is well known when all the factors have two levels: the second design is equal to the product of the first design by -1. This is called a *complete foldover* as all signs are changed. Table 19 provides a classic example, for seven 2-level factors A, B, C, D, E, F, G. If a supplementary factor N with level 1 on the first design and -1 on the second is added, the global design keeps resolution 4 with the eight factors.

The property of shifting to resolution 4 through duplication of the design in the *case of two-level factors* is real and easily proved in a much more general context than considered in this instruction manual. Any design of resolution 3 for 2-level factors, such as for instance the design of Plackett and Burman for 11 factors and 12 units derived from the 12th-order Hadamard matrix, provides, through duplication by the opposite design, a design of resolution 4, that is to say, a design in which

	Initial design (N=1)									Complement : $-1\times$ initial design ($N=-1$)							
	N	A	B	C	D	E	F	G		N	A	B	C	D	E	F	G
1	1	1	1	1	1	1	1	1	9	−1	−1	−1	−1	−1	−1	−1	−1
2	1	1	1	−1	1	−1	−1	−1	10	−1	−1	−1	1	−1	1	1	1
3	1	1	−1	1	−1	1	−1	−1	11	−1	−1	1	−1	1	−1	1	1
4	1	1	−1	−1	−1	−1	1	1	12	−1	−1	1	1	1	1	−1	−1
5	1	−1	1	1	−1	−1	1	−1	13	−1	1	−1	−1	1	1	−1	1
6	1	−1	1	−1	−1	1	−1	1	14	−1	1	−1	1	1	−1	1	−1
7	1	−1	−1	1	1	−1	−1	1	15	−1	1	1	−1	−1	1	1	−1
8	1	−1	−1	−1	1	1	1	−1	16	−1	1	1	1	−1	−1	−1	1

Table 19: 2^{8-4} of resolution 4 obtained by foldover of a 2^{7-4} of resolution 3

the common estimation of main effects as the half-difference of the means between levels 1 and -1 is unbiased even if certain couples of factors display interactions.

If the initial design is a regular design that can be constructed by the methods considered here, the same applies to the duplicated design. Thus in the example of Table 19, the defining relations of the initial design with 8 units are $D = AB$, $E = AC$, $F = BC$, $G = ABC$. It is then automatically the case, if N is the pseudofactor taking the value 1 for the first design, -1 for the second, that the global design is defined from the 4 basic factors N, A, B, C by the relations $D = ABN$, $E = ACN$, $F = BCN$, $G = ABC$. It is quite apparent that the defining relations of the initial design which overall remain valid are relations involving an even number of letters A to G. In particular, three-letter relations verified by the initial design are no longer verified overall. For example, relation $ABD = 1$ verified for the initial design cannot be valid overall since $ABD = -1$ for the additional design. The defining relations of the overall design therefore include a minimum of 4 letters and this is indeed a design of resolution 4. Note that this resolution is kept if the supplementary factor N is taken into account.

When some factors A, B, ... have four levels, this same process of duplication by the opposite design, applied to the 2-level factors and to pseudofactors A_1, A_2, B_1, B_2, ... derived from the 4-level factors, does not lead to a design of resolution 4 in every case. In order to produce such a design, the initial design of resolution 3 must not have any defining word with three factors involving an even number of symbols.

For instance assume there are three 4-level factors A, B, C decomposed into pseudo-factors A_1, A_2, B_1, B_2, C_1, C_2, and that D, E are 2-level factors. Then products such as A_1A_2DE, $A_1A_2B_1B_2C_1C_2$ must be different from 1 in the initial design. Otherwise these products remain equal to 1 in the opposite design and give rise to defining relation with only 3 factors in the overall design. The resolution of the latter then does not exceed 3 since it make an interaction of three factors confounded with the mean.

One way to avoid these defining words with three factors but an even number of symbols is to introduce two pairs {*model, part to be estimated*} in the search of the initial design, as illustrated in table 20 for the case of a design with 32 units only. In this table $q = p + A + B + C = A + B + C + D + E + \cdots + M$ is the sum of all factors. The requirement imposed by pair 1 is that q is estimable in model q which is equivalent to resolution 3. The second pair has an empty model and its *"part to estimate"*, $s.s.s + s.r.r$,

includes all products involving three factors but whose sign is unchanged when all the pseudofactors A_1, ..., C_2 and two-level factors D, E, F, ..., have their sign changed.

Indeed, note that when developed, $s.s.s$ includes, alongside with $A_1A_2B_1B_2C_1C_2$, terms such as $A_1A_2A_1A_2A_1A_2$ or $A_1A_2A_1A_2B_1B_2$ which are respectively replaced by A_1A_2, $A_1A_2B_1B_2$. But since the first pair $\{q,q\}$ implies resolution 3, its presence already implies that these latter products cannot be constantly 1 (or -1). So the only new constraint imposed by the presence of $s.s.s$ is that the product $A_1A_2B_1B_2C_1C_2$ is not constant, but equal to 1 or -1. Of course it is possible to replace $s.s.s$ by the latter product in this case with three 4-level factors. But when there are strictly more than three 4-level factors, the product $s.s.s$ is still valid and more easy to type than the sum of the products similar to $A_1A_2B_1B_2C_1C_2$.

Then, the products in $s.r.r$ do not change sign because in each of them, the component coming from s does not change sign while both components coming from r change sign.It is not difficult to see that any product involving three factors either belongs to the above sum $s.s.s + s.r.r$ or is a product in $s.s.r + r.r.r$ which change sign on the foldover part.

As noted in section 5.4.6.2, a model is systematically completed by the terms included in the terms that appear in it. Introducing $s.s.s$ in a "model" would impose useless constraint, for instance that a product like $A_1A_2B_1B_2C_1$ with an odd number of symbols is not constant. If $s.s.s$ is among the "part to be estimated", it is not completed in the same way and does not include the above product.

The Margolin rule provides a maximum to the number of 2-level factors that can be introduced. Let indeed n_4 and n_2 be the number of 4 and 2 level factors in the initial resolution 3 design and N its number of units. Assume the design obtained by the duplication process is of resolution 4. The factor equal to 1 on this design, to -1 on the duplicated part can be added without loosing this resolution 4. Hence the duplication leads to a resolution 4 design with $2N$ units, n_4 four-level and $n_2 + 1$ two-level factors. The Margolin rule then gives

$$1 + 3n_4 + (n_2 + 1) + 9(n_4 - 1) + 3(n_2 + 1) \leq 2N$$

that is

$$n_2 \leq \frac{N}{2} + 1 - 3n_4 \ .$$

```
> par5522.key <- planor.designkey(
+   factors=c("A1","A2","B1","B2","C1","C2",LETTERS[4:13]),
+   nlevels=rep(2,16),
+   listofmodels=list(
+     c(~A1*A2+B1*B2+C1*C2+D+E+F+G+H+I+J+K+L+M,
+       ~A1*A2+B1*B2+C1*C2+D+E+F+G+H+I+J+K+L+M),
+     c(~1,
+       ~(A1:A2+B1:B2+C1:C2)^3 +
+         (A1:A2+B1:B2+C1:C2)*(A1+A2+B1+B2+C1+C2+D+E+F+G+H+I+J+K+L+M)^2)),
+   nunits=32, base=~A1+A2+B1+B2+D)
```

It is possible to replace the 2nd pair *Model, part to be estimated* by two pairs: ($r.r$,s) and ($\emptyset$,$s.s.s$). The first forbids a term in s to be equal to a product in the model $r.r$.

nb. of 4 lev. fact.	nb. of 2 lev. fact.	
	reached maximum	Margolin maximum
1	14	14
2	11	11
3	6	8
4	3	5

Table 20: Fractions $4^{n_4}2^{n_2}$ of resolution 3 and size 32, that can be duplicated in resolution 4

In the case $N = 32$ (Table 20), the backtrack search quickly ends with a failure indicating the maximum attainable n_2. In fact, this maximum is the same as that given by the Margolin rule when $n_4 \leq 2$, and it is slightly smaller when $n_4 \geq 3$. These maxima are indicated in the array at the bottom of the table.

It must be remarked that the way to select the basic factors may introduce a supplementary constraint. To avoid such a constraint, it is always possible to introduce pseudofactors that do not appear in the models or parts to be estimated, as basic factors . The search is then over all possible regular designs. The drawback is that this makes the search much more longer.

In that case, putting the four-level factors A and B among the basic factors does not introduce any constraint as there cannot be any defining word involving these two factors only. It is then possible, to avoid any constraint due the choice of basic factors, to introduce a supplementary two-level pseudofactor Z, that does not appear in the models and parts to be estimated, as the last basic factor. But in that case it can also be seen that any of the two-level factors searched for can be selected as basic one without restricting the search. This is so because, as can be easily checked by using the program with 16 units, no more than 3 two-level factors can be defined as products between the pseudofactors A_1, A_2, B_1, B_2 with the imposed constraints. Thus as soon as we are looking for four two-level factors, it is possible to choose one of them as a basic one.

Table 21 illustrates the search for the example when $N = 64$. Since the search may be very long in that case, it is stopped after a few minutes. It gives again the same maximum n_2 as the Margolin rule if $n_4 \leq 2$, and a smaller one when $n_4 \geq 3$. In the latter case, several random links (RL) were used to start the search and the result given in the bottom of the table is the better one. The definition of added factors in the case $n_4 = 4$, $n_2 = 17$ is given explicitly.

Table 22 providing the Margolin maximum and the attainable maximum when $N = 16$ is also given for the sake of completeness. The search shows the impossibility to introduce $n_4 = 3$ four-level factors in that case.

As indicated in section 4.1.2, when $n_4 = 1$, Margolin [23] provides a general way to deduce from an Hadamard matrix of order N a resolution 4 fraction with the maximum number of 2-level factors given by the Margolin rule. Agrawal and Dey [1] do the same in the case $n_4 = 2$. The number of units of the provided resolution 4 designs are $4N$ when $n_4 = 1$, $8N$ when $n_4 = 2$. Since Hadamard matrices exist for almost all N multiple of 4, i.e. of the form $N = 4k$, this gives $16k$ units when $n_4 = 1$, $32k$ units when $n_4 = 2$.

The Agrawal and Dey construct for $n_4 = 2$ appears to use a foldover of a resolution 3. This is not true for the Margolin construct for $n_4 = 1$. But by slightly modifying it, as indicated below in the case $N = 12$, it is possible to get a design with the same properties built by foldover of a resolution 3 fraction. These constructs therefore also provide resolution 3 fractions that can be duplicated in resolution 4, with the maximum number of 2-level factors, for the cases $n_4 = 1$, $16k$ units and $n_4 = 2$, $32k$ units. Since k can be any integer, these fractions exist for a number of units which are not power of 2, and they can be used in the same manner as in the regular case, that is to go on with the experiment after completing the resolution 3 design if too many factors are found to be possibly active.

A 4×2^{10} of resolution 3 with 24 units that can be duplicated by foldover in a resolution 4 design.

Let H be the classical 12×12 Hadamard matrix with a first column of 1. Let H_1 be deduced from H by multiplying by 3 the second column of H. Then the matrix $\begin{bmatrix} H \\ H_1 \end{bmatrix}$ gives the searched design. Its second column is the 4-level factor and columns 3 to 12 gives the 10 two-level factors. This number 10 is immediately found to be the Margolin maximum in that case. The duplicated part in the foldover may be obtained by changing all the level signs including those $\{-3, -1, 1, 3\}$ of the four-level factor.

```
> par5522n64.key <- planor.designkey(
+   factors=c("A1","A2","B1","B2","C1","C2","D1","D2",LETTERS[5:23]),
+   nlevels=rep(2,27),
+   listofmodels=list(
+     c(~A1*A2+B1*B2+C1*C2+D1*D2+E+F+G+H+I+J+K+L+M+N+O+P+Q+R+S+T+U+V+W,
+       ~A1*A2+B1*B2+C1*C2+D1*D2+E+F+G+H+I+J+K+L+M+N+O+P+Q+R+S+T+U+V+W),
+     c(~1,
+       ~(A1:A2+B1:B2+C1:C2+D1:D2)^3 +
+         (A1:A2+B1:B2+C1:C2+D1:D2)*
+         (A1+A2+B1+B2+C1+C2+D1+D2+E+F+G+H+I+
+                                   J+K+L+M+N+O+P+Q+R+S+T+U+V+W)^2)),
+   nunits=64, base=~A1+A2+B1+B2+C1+C2)
```

4.1.3.2 Partial foldover Instead of changing the signs of all factors, it is possible to change only some of them to get the second follow-up design, then called a *partial foldover* design *on* these factors (the sign of which is changed). Montgomery [25] studies for instance a foldover on only one factor in a classical resolution 2^{n-k}. This change gives a foldover design which when combined with the initial fraction allows the estimation of the main effect of this factor and of all its interactions with each of the other factors.

Ankenman [2] gives tables of resolution 3 fraction for mixtures of two- and four-level factors, which may be duplicated by partial foldover to get a combined design of resolution 4. We examine below how to obtain such fractions with `PLANOR` when the factors on which the foldover is performed are determined.

nb. of 4 lev. fact.	nb. of 2 lev. fact.		
	reached maximum	RL	Margolin maximum
1	30		30
2	27		27
3	22	0	24
4	17	888888	21

Solution for
$n_4 = 4, \quad n_2 = 17$

$D_1 = B_2C_2, \quad D_2 = A_1B_1B_2C_1, \quad E = A_2B_1C_1, \quad F = A_1B_2C_1, \quad G = A_2B_1C_1C_2,$
$H = A_2B_2, \quad I = A_1B_2C_1C_2, \quad J = A_1A_2B_2C_1, \quad K = A_2B_2C_2, \quad L = A_1A_2B_1C_2,$
$M = A_2B_1B_2C_1C_2, \quad N = B_1B_2C_1C_2, \quad O = A_1A_2B_2C_1C_2, \quad P = A_1A_2C_1C_2,$
$Q = A_1B_1, \quad R = A_1A_2B_1B_2, \quad S = A_1A_2B_1B_2C_2, \quad T = B_1C_1, \quad U = A_2C_2$

Table 21: Fraction $4^{n_4}2^{n_2}$ of resolution 3 and size 64, that can be duplicated in resolution 4

nb. of 4 lev. fact.	nb. of 2 lev. fact.	
	reached maximum	Margolin maximum
1	6	6
2	3	3

Table 22: Fraction $4^{n_4}2^{n_2}$ of resolution 3 and size 16, that can be duplicated in resolution 4

Consider first a four-level factor A, decomposed into pseudofactors A_1, A_2. If there is a change of sign in the foldover on one of these pseudofactors, say A_1, it is always possible to assume there is also a change of sign for the second A_2. Otherwise the product A_1A_2 also change sign and by selecting it as the second pseudofactor, that is by interchanging A_2 and A_1A_2, we return to the situation where both pseudofactors change sign. So we now assume that for each four-level factor, the foldover is either on both associated pseudofactors or on none of them.

The code below shows such a search for 32 units, $n_4 = 2$ four-level and $n_2 = 7$ two-level factors. It is performed first by complete foldover, then by partial foldover on C, E, F, L, as in the solution given in [2] table 8. Note that, with the present `PLANOR` R package and contrary to the original `PLANOR` software, it is necessary to declare the two-level pseudofactors explicitly because they appear explicitly in model or estimate formulae.

```
> set.seed(1234)
> completeFoldover.key <- planor.designkey(
+   factors=c("A1","A2","C1","C2",LETTERS[c(5:8,10:12)]),
+   nlevels=rep(2,11),
+   listofmodels=list(c(~A1*A2+C1*C2+E+F+G+H+J+K+L,
+                       ~A1*A2+C1*C2+E+F+G+H+J+K+L),
+                     c(~(A1+A2+C1+C2+E+F+G+H+J+K+L)^2,
+                       ~A1:A2+C1:C2)),
+   nunits=32, base=~A1+A2+C1+C2+E, max.sol=10, randomsearch=TRUE)
```

```
Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 6 to 11
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
      first visit to column 10
      first visit to column 11
The search is closed: max.sol =  10 solution(s) found
```

```
> partialFoldover.key <- planor.designkey(
+   factors=c("A1","A2","C1","C2",LETTERS[c(5:8,10:12)]),
+   nlevels=rep(2,11),
+   listofmodels=list(
+     c(~A1*A2+C1*C2+E+F+G+H+J+K+L,
+       ~A1*A2+C1*C2+E+F+G+H+J+K+L),
+     c(~1, ~(A1*A2+G+H+J+K+C1:C2)^3 +
+       (A1*A2+G+H+J+K+C1:C2)*(C1+C2+E+F+L)^2)),
+   nunits=32, base=~A1+A2+C1+C2+E, max.sol=10, randomsearch=TRUE)
```

```
Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 6 to 11
      first visit to column 6
      first visit to column 7
      first visit to column 8
      first visit to column 9
      first visit to column 10
      first visit to column 11
The search is closed: max.sol =  10 solution(s) found
```

In the complete foldover search, the first pair *model, part to be estimated* is the requirement for obtaining resolution 3. Then to obtain a design that produces a resolution 4 design when duplicating it by partial foldover on the four level factor C (that is on C_1 and C_2) and on the two-level factors E, F, L, it is necessary to ensure that no defining product involving three factors keeps the same sign on the foldover. This is done by the second pair *model, part to be estimated* with an empty model. The part to be estimated $q.q.q + q.r.r$ involves two model parts q and r. The first q includes all factors or pseudofactors that do not change sign. Among them are A_1, A_2, $A_1.A_2$ and $C_1.C_2$. On the contrary, the model part r includes those factors or pseudofactors that change sign. The products in $q.q.q + q.r.r$ are therefore those involving three factors that do not change sign. Since these products appear in the "part to be estimated" associated with an empty model, they cannot be constant on the fraction.

One way of comparing the 10 designs obtained by this search is to look at their *word length pattern* giving for each pair (m_2, m_4) the number $W(m_2, m_4)$ of defining word involving m_2 two-level and m_4 four-level factors. Since confounding between main effects and two-factor interactions arises from defining word of length 3, a possible criterium to select the design would be to minimise the number of defining word with three factors, that is $L(3) = W(0,3) + W(1,2) + W(2,1) + W(3,0)$. In the complete foldover (top of table 23), this minimum is 5. But in fact, an analysis of the aliases in the model $p.p$ which contains all two factor interactions shows that while only 3 degrees of freedom of the main effects are estimable with this minimum of 5, there is a solution with $L(3) = 8$ which allows to estimate up to 4 degrees of freedom of the main effects. In the partial foldover (bottom of table 23) this minimum is 4 and the corresponding solution is that which allows to estimate the maximum number 4 of degrees of freedom of the main effects.

The second important thing to consider is then the properties of the combined resolution 4 designs. The only way to study their properties is to introduce the factor definitions manually using the field "Predetermined factors". As already explained, they can be easily obtained from the definitions of the added factors in the initial fraction. A basic pseudofactor Z taking level 1 on the fraction, -1 on the foldover, is introduced. Then every definition remaining true after the change of sign is kept, while if the signs of the added factor and of the product defining it differ after the change, the pseudofactor Z is simply added in the definition.

Again criteria that may be used to compare the combined designs are the number $L(4)$ of words of length 4, the number of degrees of freedom of the two-factor interactions that can be estimated in a model with all interactions, and finally the number of sets of 2, 3, ... aliased two-factor interactions. We give these criteria in table 23, first for the ten designs that can be duplicated by complete foldover randomly obtained by `PLANOR` , then for the ten that can be duplicated by partial foldover also randomly obtained as well as for the design proposed by Ankelman.

If partial foldover is used in the frame of a model including all two-factor interactions, Ankelman's solution appears to give the maximum 4 of estimable main effects in the initial fraction, the maximum 21 of estimable interactions in the combined design, finally the smaller number 3 of groups of strictly more than two confounded interactions. This thus appears to be the best solution and can of course be introduced in `PLANOR` using the field with the predetermined factors in order to get it explicitly.

But it must be observed that solution 2 in the search for a complete foldover allows to estimate 32 degrees of freedom of the interactions, that is 11 more than any of the other fractions, when considering the combined design. Though the initial resolution 3 design in that case does not allow the unbiased estimation of any main effect in the model with all two-factor interactions, that property of the combined design should make this fraction more attractive than the others in some situation.

4.1.4 Introduction of blocks

The introduction of blocks and of the hierarchical constraint for temperature in the example considered in section 4.1.1 is achieved without any difficulty.

That can be duplicated by complete foldover

fraction	1	2	3	4	5	6	7	8	9	10
(1)	5	6	7	5	7	8	7	7	6	7
(2)	3	0	0	3	1	4	1	2	3	2
(3)	13	16	11	14	13	14	13	11	13	14
(4)	18	32	18	20	23	20	23	18	18	20
(5)	8	7	4	9	8	9	8	4	8	9

That can be duplicated by foldover on C, E, F, L

fraction	1	2	3	4	5	6	7	8	9	10	$*$
(1)	4	6	6	7	7	8	8	6	6	7	4
(2)	4	3	3	2	2	2	2	1	1	2	4
(3)	12	11	11	12	12	14	14	12	10	12	10
(4)	17	21	21	18	18	14	14	18	21	18	21
(5)	4	5	5	5	4	7	7	6	3	6	3

Table 23: Properties of some fractions for 32 units that can be duplicated

The fraction are for $n_4 = 2$ four-level, $n_2 = 7$ two-level factors. They can be duplicated in resolution 4 by complete foldover (first array) or by partial foldover (second array). Row (1) and (2) are relative to the initial fraction, rows (3) (4), (5) to the combined design with 64 units.

$*$: Ankelman's fraction

(1) : number $L(3)$ of defining word of length 3 in the initial fraction

(2) : number of main effects of the initial fraction that are estimable in the model including all two-factor interactions

(3) : number $L(4)$ of defining word of length 4 in the combined design.

(4) : number of interactions estimable in the combined design

(5) : number of sets of 3 or more aliased interactions in the combined design.

The code below specifies the parameters of the search for a design of resolution 4, size 64, that includes 3 factors with 4 levels and 7 factors with 2 levels, including temperature. Note that the order in the `factors` argument of `planor.designkey` may have an influence on the speed of the search. Here, it is more efficient to declare Bl and T before the other treatments factors.

```
> iblock.key <- planor.designkey(factors=c("Bl","T",LETTERS[1:9]),
+                                nlevels=c(8,2,4,4,4,rep(2,6)),
+                                block=~Bl,
+                                hierarchy=~T/Bl,
+                                listofmodels=list(
+                                  c(~Bl + (A+B+C+D+E+F+G+H+I+T)^2,
+                                    ~A+B+C+D+E+F+G+H+I),
+                                  c(~(A+B+C+D+E+F+G+H+I+T)^2, ~T)),
+                                nunits=64, base=~A+B+C,
+                                max.sol=20, randomsearch=TRUE)
```

Design definition, solution 6, random link 520571533

key matrix

blocs	$\star$ Bl_1	$\star$ Bl_2	$\star$ Bl_3	T	D	E	F	G	H	I
A_1	0	1	0	1	1	0	1	0	1	0
A_2	1	1	0	0	1	1	1	1	0	1
B_1	0	1	1	0	1	1	1	1	1	0
B_2	0	0	1	1	0	1	1	0	0	1
C_1	1	1	1	1	0	0	1	1	1	1
C_2	0	0	0	0	1	1	0	0	1	1

defining relations

Bl_1	=	A_2C_1
Bl_2	=	$A_1A_2B_1C_1$
Bl_3	=	$B_1B_2C_1$
T	=	$A_1B_2C_1$
D	=	$A_1A_2B_1C_2$
E	=	$A_2B_1B_2C_2$
F	=	$A_1A_2B_1B_2C_1$
G	=	$A_2B_1C_1$
H	=	$A_1B_1C_1C_2$
I	=	$A_2B_2C_1C_2$

Table 24: Resolution 4 fraction of a $4^3 \times 2^7$ in 8 blocks of 8 units

Among the 20 solutions obtained, the selected one provides a maximum number of 26 unaliased effects in the model including the block effect and the two-factor interactions. These unaliased effects, which include the $15 = 3\times3+6$ main effects of the factors differing from temperature T, appear in Table 25, which also provides the 37 sets of aliased effects. Once the $63 = 26 + 37$ unaliased effects and linear combinations of aliased effects have been estimated, there are no degrees of freedom left to estimate error variance. Similarly, all the block effects are observed to be confounded with interactions. It is therefore not possible to identify degrees of freedom for the estimation of *inter-block* variance against which the main effect of the temperature factor T confounded with the blocks is normally tested.

In fact, neither do the 19 other solutions obtained by this search allow to identify degrees of freedom for the estimation of inter- or intra-block variances. With this type of design which contains a maximum number of 2-level factors, we are thus compelled to perform the analysis using techniques such as those described and referenced in [18], [20], with the additional problem caused by the presence of two strata, the inter- and the intra-block ones.

detailed list of the sets of aliased effects in the model	
$[Bl_2]; C_1C_2D; B_2F; A_1G;$ $[Bl_3]; B_1B_2C_1; A_1A_2F;$ $[Bl_2Bl_3]; A_1A_2B_2;$ $[Bl_1]; DH; A_2C_1; B_1G;$ $[Bl_1Bl_2]; A_1B_1; C_1C_2H;$ $[Bl_1Bl_3]; C_2E; A_2B_1B_2;$ $[Bl_1Bl_2Bl_3]; T;$ $DI; B_1T; A_2F;$ $C_1C_2I; A_2B_2;$ $B_1B_2F; A_1A_2C_1;$ $C_1F; HI; A_1A_2B_1B_2; TG;$ $A_1T; B_2C_1;$ $B_1B_2T; C_2H; A_1A_2G;$ $C_1C_2E; A_1F; B_2G;$ $DE; C_1T; A_1B_2; FG;$ $A_1A_2B_1; C_2D;$ $A_2T; B_1F; EH;$ $EI; B_1C_1; A_2G;$	$A_1C_1; B_2T;$ $A_2B_1; TF; C_1G;$ $A_1A_2T; C_2I; B_1B_2G;$ $TE; C_1D; A_2H;$ $B_2I; A_2C_1C_2;$ $FI; C_1H; A_2D;$ $C_2T; B_1B_2H; A_1A_2I;$ $B_1B_2I; A_1A_2H; C_2G;$ $B_1E; C_1I; FH;$ $B_1C_2; A_1A_2D;$ $A_1E; C_1C_2F; B_2D;$ $DF; B_2C_1C_2; A_2I; EG;$ $A_2C_2; B_1B_2E;$ $TD; B_1I; C_1E;$ $A_2E; B_1B_2C_2; TH; GI;$ $A_1A_2C_2; TI; B_1D; GH;$ $B_1C_1C_2; A_1H;$ $A_1C_1C_2; EF; B_1H; DG;$ $A_1D; B_2E; C_1C_2G;$
list of unaliased effects	
$A_1; A_2; A_1A_2; B_1; B_2; B_1B_2; C_1; C_2; C_1C_2; D; E; F; G; H; I;$ $> A_1B_1B_2; A_1C_2; A_1A_2C_1C_2; B_2C_2; B_1B_2C_1C_2; B_1B_2D;$ $> B_2H; C_1C_2T; A_1I; C_2F; A_1A_2E;$	
list of unaliased block effects (empty)	

Table 25: Alias structure for the design of table 24

An alternative is to reduce the number of 2-level factors and then attempt to identify degrees of freedom to estimate the two errors. For inter-block error, it is known that one of the block effects is confounded with temperature. It might be sought to ensure that the six other block effects be unaliased, but it is easily demonstrated that when there are three four-level factors A, B, C, at least one effect of each of the interactions AB, AC, BC is confounded with the blocks.

Let us consider, for instance, interaction BC. In the products defining the seven block effects in relation to the basic pseudofactors, A can only appear in the three forms A_1, A_2, A_1A_2. If it appears in strictly more than 3 of these products, it necessarily appears in the same form in two of them and consequently does not appear in the product of these two. Thus there is at least one product defining a block effect where only B and C appear and clearly both must appear since a block effect cannot be confounded with a main effect.

There are therefore in this case at least four aliased block effects and at most three unaliased block effects. In order to be sure to obtain a design providing the maximum number of unaliased block effects, three block effects are added to the *part to be estimated* number 1 of table 24, either in the form Bl_1, Bl_2, $Bl_1.Bl_2$ of three linked effects, or in the form Bl_1, Bl_2, Bl_3 of three independent effects (tab. 26). Note that to indicate an index to `PLANOR` , one uses the underline symbol, that is Bl_1 for Bl_1, Bl_2 for Bl_2.

	model		part to be estimated
1	Bl+S.S	or	Bl_1+Bl_2+$Bl_1.Bl_2$+S1 Bl_1+Bl_2+Bl_3+S1
2	S.S		T

Table 26: Modif. of the search in tab. 24 in order to estimate the inter-block variance

In both cases, the first 4 two-level factors, among which are T, are obtained very rapidly and the search dwells on the 5th factor. Thus four 2-level factors are selected in addition to the three 4-level factors and the 8-level block factor.

Then for each of the two possible choices for the "part to be estimated" number 1, twenty solutions are searched with a non-zero random link.

When the "part to be estimated 1" are $Bl_1+Bl_2+Bl_1.Bl_2$, the analysis of aliases for the 20 solutions reveals the same alias structure (which leads to the idea that there is in fact only one solution, allowing for possible permutations of factors or pseudofactors). This alias structure includes 26 unaliased effects and 32 sets of aliased effects. It thus leaves $5 = 63 - (26 + 32)$ degrees of freedom to estimate residual variance and by construction contains three degrees of freedom to estimate inter-block variance.

When the "part to be estimated 1" are $Bl_1+Bl_2+Bl_3$, the analysis of aliases for the 20 solutions reveals several structures. In addition to the three unaliased block effects, these structures generally include 32 or 33 unaliased treatment effects, 24 or 25 sets of aliased effects and leave 3 or 4 degrees of freedom to estimate residual variance. Nevertheless, one

of these, solution 2, is clearly distinguishable from the 19 others: it includes 42 unaliased treatment effects and 17 sets of effects. It thus leaves only 1 degree of freedom for the residual, but appears clearly superior to the other solutions. Tables 27 and 28 specify this solution and provide its alias structure.

Model parts	Hierarchies
1 S1:A+B+C+D+E+F	1 T : Bl
2 S :S1+T	
Models	**Part to be estimated**
1 Bl+S.S	1 Bl_1+Bl_2+Bl_3+S1
2 S.S	2 T

Design HYGIEN2, solution 2, random link 1691852795

key matrix

blocs	*	*	*				
	Bl_1	Bl_2	Bl_3	T	D	E	F
A_1	1	1	1	1	0	1	0
A_2	1	0	1	0	1	1	1
B_1	0	0	1	1	0	1	1
B_2	1	1	0	0	1	1	1
C_1	1	0	0	1	1	1	0
C_2	1	1	1	1	0	0	1

defining relations

$Bl_1 = A_1A_2B_2C_1C_2$
$Bl_2 = A_1B_2C_2$
$Bl_3 = A_1A_2B_1C_2$
$T = A_1B_1C_1C_2$
$D = A_2B_2C_1$
$E = A_1A_2B_1B_2C_1$
$F = A_2B_1B_2C_2$

Table 27: Resolution 4 fraction of a $4^3 \times 2^4$ in 8 blocks of 8 units

Of course, there is nothing to preclude duplication of a block in such a design This provides an additional degree of freedom for inter-block residual variance and 7 for intra-block residual variance. These degrees of freedom provide "*pure*" error variances insofar as they are not inflated by potential interactions of three or more factors. Nevertheless, the statistical analysis of the non-regular design thus obtained is more complex and the practical relevance of this kind of repetition appears to be of little relevance, in the context of the screening of influencing factors, in which the absence of effects of several factors provides the possibility, on the contrary, of identifying the active factors without difficulty.

Taking into account blocks in designs of resolution 3 as those considered in tables 20, 21, 22 may be done by requiring that the main effects different from the temperature effect are estimable within the model including all main effects and the block effect. In the considered cases, the allocation among blocks of size 8 appears possible without any reduction of the number of two-level factors. When duplicating, one add as supplementary block pseudofactor the one equal to 1 on the first part, to -1 on the duplicated part. Table 29 shows for instance how to introduce the block search in the case with $N = 32$ units, $n_4 = 2$ four-level, $n_2 = 11$ two-level factors. The symbol $\sim$ is used in this example to substract T from the part to be estimated. Note that if T was not to be constant on each block and was to be estimated within blocks, the number of two-level factors would have in that example to be reduced of 1.

Through the study of aliases (section 5.6), one gets the *"word pattern"* for each of the 11 solutions randomly obtained by the backtrack search defined in table 29. This word

detailed list of sets of aliased effects in model $Bl + S.S$	
$[Bl_1Bl_2Bl_3]; T;$ $[Bl_1Bl_3]; B_1B_2C_1; A_1A_2E;$ $[Bl_2Bl_3]; C_2F; A_2B_1B_2;$ $[Bl_1Bl_2]; A_2C_1; B_2D;$ $A_1A_2B_1B_2; C_1E;$ $A_1A_2C_1; B_1B_2E;$ $B_2C_1; A_2D;$ $A_2B_2; C_1D;$	$B_1B_2F; A_2C_2;$ $B_1B_2C_2; A_2F;$ $A_1T; B_1C_1C_2; DF;$ $C_1C_2F; A_1E; B_1D;$ $B_1F; TE; C_1C_2D;$ $B_1T; EF; A_1C_1C_2;$ $A_1B_1; C_1C_2T; DE;$ $C_1C_2E; A_1F; TD;$ $TF; B_1E; A_1D;$
list of unaliased treatment effects	
$A_1; A_2; A_1A_2; C_1; C_2; C_1C_2; B_1; B_2; B_1B_2; D; E; F;$ $A_1B_2; A_1B_1B_2; A_2B_1; A_1A_2B_1; A_1A_2B_2; A_1C_1;$ $A_1C_2; A_2C_1C_2; A_1A_2C_2; A_1A_2C_1C_2; B_1C_1; B_1C_2;$ $B_2C_2; B_2C_1C_2; B_1B_2C_1C_2; A_1A_2D; A_2E; A_1A_2F;$ $A_2T; A_1A_2T; B_1B_2D; B_2E; B_2F; B_2T; B_1B_2T; C_2D;$ $C_2E; C_1F; C_1T; C_2T;$	
list of unaliased block effects	
$Bl_1; Bl_2; Bl_3;$	

Table 28: Alias structure for design of table 27

pattern gives in this case, for each couple (m_2, m_4) of integer, the number of defining words including m_2 two-level, m_4 four-level factors. Fractions that have different word patterns cannot be equivalent in the sense there is no permutation of the factors transforming one into the other. Fractions that have the same word pattern may be non equivalent, but there is a high probability there are in fact equivalent.

The 11 solutions have in fact three different word patterns which suggests that, up to a permutation of the factors, there are three really different fractions. The solution 1, 2 and 5 are found to be representatives of these three fractions. It is then easy to find the defining contrasts of the resolution 4 fractions of size 64 obtained by duplicating them by complete foldover, and to study and compare aliasing in them. Solution 2 appears to be the one that gives the biggest number 8 of unaliased two factor interactions (versus 7 in solution 1 and 6 in solution 5). That solution 2 has 20 sets with only two aliased such interactions (versus 12 and 10), but it has 15 sets of 5 and more aliased two-factor interactions (versus 7 and 5). It is therefore not clearly better than the two other solutions, but we selected it as an example (table 30) to illustrate how the defining contrast of the duplicated resolution 4 fraction are deduced from those of the initial resolution 3 one.

The example of a search of a resolution 3 fraction with 64 units, 4 four-level and 10 two-level factors that can be duplicated in resolution 4, shows again that partial foldover may be less interesting than the complete foldover. The latter can estimate up to 37 isolated two-factor interactions, has 29 groups of two aliased two-factor interactions and only 32 of strictly more than two such interactions. The introduction of blocks in it however reduces the number of estimable two-factor interactions as six of them are aliased

	name	:	HYGIEN3.REG
	nb. of units	:	32
	Definition of basic factors	:	user
	factor decomposition type	:	maximum
⋆	Backtrack search - time limit	:	10 mn
	- nb. max. sol	:	11
	- random link	:	123456

Basic factors	
fac.	nb. niv.
A	4
B	4
C	2

Added factors							
fac.	nb.lev.	bloc	fac.	nb.lev.		fac.	nb.lev.
Bl	4	←	G	2		K	2
D	2		H	2		L	2
E	2		I	2		T	2
F	2		J	2			

Model parts

p : $C + D + E + F + G + H + I + J + K + L + T$
q : $p + A + B$
r : $p + A_1 + A_2 + B_1 + B_2$
s : $A_1A_2 + B_1B_2$

Model		Part to be estimated	
0	q+Bl	0	q∼ T
1	q	1	T
2	r.r	2	s

Hierarchies

1 T : Bl

Table 29: Blocking a fraction $4^2 2^{11}$ of resolution 3 and size 32 that can be duplicated in resolution 4

2nd key matrix (random link 1311708280)

blocs						*	*									
	A_1	A_2	B_1	B_2	C	Bl_1	Bl_2	D	E	F	G	H	I	J	K	L	T
A_1	1	0	0	0	0	1	1	1	1	0	0	1	0	1	1	0	1
A_2	0	1	0	0	0	0	1	1	1	1	1	0	1	1	0	0	0
B_1	0	0	1	0	0	1	0	1	0	0	1	0	0	1	1	1	1
B_2	0	0	0	1	0	0	1	1	1	1	1	0	1	1	0	0	0
C	0	0	0	0	1	0	0	0	1	1	1	1	0	1	1	1	0

Defining relations

of resol. 3 fraction (multiplicative)			Predetermined factors of its resol. 4 complete foldover			
Bl_1	=	A_1B_1	1	Bl_1	=	A_1B_1
Bl_2	=	$A_1A_2B_2$	2	Bl_2	=	$A_1A_2B_2$
D	=	$A_1A_2B_1B_2$	3	D	=	$A_1A_2B_1B_2Bl_0$
E	=	$A_1A_2B_2C$	4	E	=	$A_1A_2B_2CBl_0$
F	=	A_2B_2C	5	F	=	A_2B_2C
G	=	$A_2B_1B_2C$	6	G	=	$A_2B_1B_2CBl_0$
H	=	A_1C	7	H	=	A_1CBl_0
I	=	A_2B_2	8	I	=	$A_2B_2Bl_0$
J	=	$A_1A_2B_1B_2C$	9	J	=	$A_1A_2B_1B_2C$
K	=	A_1B_1C	10	K	=	A_1B_1C
L	=	B_1C	11	L	=	B_1CBl_0
T	=	A_1B_1	12	T	=	$A_1B_1Bl_0$

Table 30: Solution 2 in the backtrack search of table 29

with block effects in the result of the search.

4.1.5 About the notion of minimum aberration

Minimum aberration has been introduced by Fries and Hunter [13] as a tool to select a fraction among those a a given resolution. As pointed out in [18], it is only when the number of words of minimum length is small that a design with minimum aberration will make estimable the maximum number of effects. For instance in the case of a regular resolution 4 fraction 2^{h-m} with W_4 words of length 4, table 31 gives the minimum number k_2 of unaliased two-factor interactions. The table shows that a fraction with $W_4 = 7$ make in some case only 21 interactions confounded, while a fraction with $W_4 = 5$ (resp. $W_4 = 6$) will make at least 24 (resp. 28) two-factor interactions confounded. Indeed the minimum aberration fraction 2^{9-4} has $W_4 = 6$ defining words of length 4 and make 28 two-factor interactions confounded whereas the 2^{9-4} regular fraction appearing in the tables [26] has $W_4 = 7$, that is one more defining word of length 4, but make only 21 two-factor interactions confounded.

W_4	1	2	3	4	5	6	7	> 7
k_2	6	12	15	21	24	28	21	≥ 21

Table 31: Minimum number of aliased two-factor interaction in resolution 4

4.2 Mixture of quantitative and qualitative factors

4.2.1 Polynomial and pseudofactorial effects

Let us now assume that at least one of the 4-level factors, for example A, is quantitative. Dissymmetry then occurs between the three associated effects A_1, A_2, A_1A_2 and it is possible to draw on this dissymetry to select the design. Here we provide an overview of the way of proceeding. A more detailed description and, in particular, the precise definition of polynomial effects, can be found in [18], [7].

By *polynomial effects* we here refer to all the effects, regardless of the qualitative or quantitative nature of the factors they contain, where the quantitative factors appear in polynomial form. For instance, if?A is quantitative with 4 levels, and B is qualitative with 2 levels, the polynomial effects are the main effects B, $\mathrm{lin}\, A$, $\mathrm{quad}\, A$, $\mathrm{cub}\, A$ and interactions $\mathrm{lin}\, A.B$, $\mathrm{quad}\, A.B$, $\mathrm{cub}\, A.B$. They should be distinguished from the *pseudo-factorial effects* associated with the products of pseudofactors: B, A_1, A_2, A_1A_2, $A_1.B$, $A_2.B$, $A_1A_2.B$. When there are quantitative factors, it is the polynomial effects which are really meaningful, formalize hypotheses and must be estimated.

The construction method used by `PLANOR` is not directly adapted to quantitative factors and to the estimation of polynomial effects. Nevertheless, if the correspondence between the quantitative levels and pseudofactor levels is chosen appropriately, the pseudo-factorial effects are expressed in relation to the polynomial effects in a simple form, which

may be utilized to search for defining relations between pseudofactors which are adapted to the quantitative nature of certain factors.

More specifically, for each 4-level quantitative factor, the correspondence between ordered levels and pseudofactor levels is chosen as indicated in table 32. This choice is the standard `PLANOR` choice and it is therefore not necessary to redefine the levels in order to obtain it.

The ordered levels appearing in table 32 are the first four integers 0, 1, 2, 3, which can always be reverted to by changing the origin and scale. The correspondence of table 32 produces the relations appearing in table 33.

	notation additive		notation multiplicative	
A	A_1	A_2	A_1	A_2
0	1	1	-1	-1
1	1	0	-1	1
2	0	1	1	-1
3	0	0	1	1

Table 32: Pseudofactors decomposition of a quantitative factor A

A_1	=	$(2\,\mathrm{lin}\,A - \mathrm{cub}\,A)/\sqrt{5}$	$\mathrm{lin}\,A$	=	$(2A_1 + A_2)/\sqrt{5}$
A_1A_2	=	$\mathrm{quad}\,A$	$\mathrm{quad}\,A$	=	A_1A_2
A_2	=	$(\mathrm{lin}\,A + 2\,\mathrm{cub}\,A)/\sqrt{5}$	$\mathrm{cub}\,A$	=	$(-A_1 + 2A_2)/\sqrt{5}$

Table 33: Relations between polynomial and pseudofactorial main effects

These relations are used to express a pseudofactorial effect in relation to polynomial effects or, reciprocally, a polynomial effect on the basis of pseudofactorial effects. The products obtained following substitution of the terms appearing on the left-hand side of the equalities in table 33 for those appearing on the right-hand side may then be developed in the usual manner. For example, if B is also a 4-level quantitative factor decomposed in a similar fashion to A, and C is a two-level factor:

$$
\begin{aligned}
A_2C &= (\mathrm{lin}\,A + 2\,\mathrm{cub}\,A)C/\sqrt{5} = (\mathrm{lin}\,A.C + 2\,\mathrm{cub}\,A.C)/\sqrt{5}\ , \\
A_1B_1 &= (2\,\mathrm{lin}\,A - \mathrm{cub}\,A)(2\,\mathrm{lin}\,B - \mathrm{cub}\,B)/5 \\
&= (4\,\mathrm{lin}\,A\,\mathrm{lin}\,B - 2\,\mathrm{cub}\,A\,\mathrm{lin}\,B - 2\,\mathrm{lin}\,A\,\mathrm{cub}\,B + \mathrm{cub}\,A\,\mathrm{cub}\,B)/5, \\
A_1A_2B_1 &= \mathrm{quad}\,A(2\,\mathrm{lin}\,B - \mathrm{cub}\,B)/\sqrt{5} \\
&= (2\,\mathrm{quad}\,A\,\mathrm{lin}\,B - \mathrm{quad}\,A\,\mathrm{cub}\,B)/\sqrt{5} \\
\mathrm{lin}\,A\ \mathrm{lin}\,B &= (2A_1 + A_2)(2B_1 + B_2)/5 \\
&= (4A_1B_1 + 2A_2B_1 + 2A_1B_2 + A_2B_2)/5\ .
\end{aligned}
$$

These relations and the assumptions made on polynomial effects enable nesting of the different pseudofactorial effects and suggest search strategies for regular designs.

When all the factors are qualitative, the main criterion for classifying a factorial effect is the number of factors which appear. A realistic assumption, when little information is available a priori, is that the higher this number, the weaker such an effect is.

In particular, interactions of three or more factors are often assumed to be zero, which results in the search for resolution 5 designs.

This number of factors appearing in the effect remains an important element in classifying polynomial effects in cases in which there are quantitative factors. Nevertheless, another criterion is involved in this case: the degree of the polynomial effect. This degree is 1 for a qualitative factor and equal to the polynomial degree for a quantitative effect, that is to say, 1 for a linear effect, 2 for a quadratic effect, 3 for a cubic effect. The degree of an effect which involves several factors is the sum of degrees for each factor. Thus the degrees of $\mathrm{lin}\, A\, \mathrm{quad}\, B$, $\mathrm{cub}\, A.C$ are $3 = 1 + 2$ and $4 = 3 + 1$, respectively. A hypothesis often formulated is that polynomial effects of degree 3 or above are zero. A less restrictive hypothesis is that only the polynomial effects of degree 3 which do not belong to the main effects, i.e. involving at least two factors, are equal to zero.

Two approaches may be adopted to make effective use of the hypotheses on polynomial effects. The first [14] is exclusively based on pseudofactorial effects equal to zero with a view to obtaining a regular design. The resulting design is orthogonal with respect to the polynomial effect and enables estimation of polynomial effects with an efficiency of 1, that is to say, with a variance identical – allowing for size adjustment – to that of a full factorial design. The second approach [8], based on a more refined use of the relations in table 33, provides designs with a much lower number of units. The latter are not orthogonal but display nonetheless excellent efficiencies in estimating polynomial effects.

Both approaches generally give rise to designs with properties of uniform distribution of points which make them quite robust with respect to the selected model and, for this reason, preferable in many cases to the D-optimal designs obtained by algorithmic procedures. The description which we provide here of these approaches is further set out in [18] which can be referred to for additional information.

4.2.2 Regular designs that are orthogonal for the polynomial model

Under the assumption that polynomial effects of degree 3 or above are zero, the pseudofactorial effects expressed only from polynomial effects of degree 3 or above are equally zero. Including the other non-zero pseudofactorial effects in the model and part to be estimated ensures that the design obtained enables estimation of all the polynomial effects of degree 1 or 2. It is moreover orthogonal with respect to these polynomial effects which it estimates with the same efficiency as a full factorial design, that is to say, with an efficiency of 1. This efficiency is not always in this case maximum efficiency, but the designs thus constructed have a very good overall efficiency and also good robustness with respect to the model.

Let us consider for instance a case in which there are 16 units, two 4-level quantitative factors A and B, and one two-level quantitative factor C.

If the polynomial effects of degree 3 or above are zero, the same applies to all the pseudofactorial effects containing 3 or more symbols. For example,

$$A_1A_2B_1 = (2\,\mathrm{quad}\, A\,\mathrm{lin}\, B - \mathrm{quad}\, A\,\mathrm{cub}\, B)/\sqrt{5}$$

is zero since quad A lin B and quad A cub B, of degrees 3 and 5, respectively, are both zero.

The design of resolution 5 defined by $A_1A_2B_1B_2C = 1$ then enables estimation of all the pseudofactorial effects with 1 or 2 symbols, and subsequently all the polynomial effects of degree 1 or 2. It is thus adapted to the quantitative nature of factors A and B. This design may be obtained in `PLANOR` by defining the model and part to be estimated by $P.P$ where P is the model part defined by $P : A\_1 + A\_2 + B\_1 + B\_2 + C$.

In general, the smallest degree of polynomial effects on the basis of which a pseudofactorial effect is expressed is easily obtained. For instance, if A, B are quantitative with 4 levels, C is qualitative with 4 levels, this smallest degree is 4 for effect $A_1B_1B_2C_1C_2$. In order to obtain it, the number of pseudofactors derived from quantitative factors and the number of qualitative factors are totalled, that is to say, 3 for the three pseudofactors A_1, B_1, B_2 and 1 for the qualitative factor C.

In accordance with this calculation method, if the design is of resolution 5 when the pseudofactors derived from quantitative factors are assimilated to factors, it enables orthogonal estimation of all the effects of a polynomial model of degree 2.

For instance, with 64 units, it is possible to study in resolution 5 up to 8 two-level factors. The replacement of certain factors by pseudofactors derived from 4-level quantitative factors results in orthogonal designs for a model of degree 2 with 1, 2, 3 or 4 four-level quantitative factors and 6, 4, 2 or 0 two-level factors, respectively. Table 34 provides an example with 2 four-level quantitative factors A, B, and 4 two-level factors C, D, E, F. A four-level block factor BL was added. Finally, the main effects of A and B were included in the model and part to be estimated in order to enable the estimation of effects cub A and cub B. It would have been possible, without restricting choice, to take A, B together with two of the 2-level factors as basic factors. However, the standard option was selected in this case to define the basic factors, in order to illustrate the corresponding outputs.

4.2.3 Non orthogonal for the polynomial model regular designs

4.2.3.1 $1/2$ **fraction of a** 4×4 **for two quantitative factors** Let us consider a case in which there are two 4-level quantitative factors A and B and a model assumed to be of degree 2. This model has 6 parameters: the constant, lin A, lin B, quad A, quad B and lin A lin B. But there are only 5 pseudofactorial effects whose expression in relation to polynomial effects does not display any of these parameters and which are therefore zero: $A_1A_2B_1$, $A_1A_2B_2$, $A_1B_1B_2$, $A_2B_1B_2$, $A_1A_2B_1B_2$. The method described in the previous paragraph therefore does not help to find an appropriate $1/2$ fraction, because such a fraction should enable the estimation of $11 = 16 - 5$ parameters with only 8 units. Since the polynomial effects of degree 3 or above are assumed to be zero, the pseudofactorial effects are expressed in the form provided in table 35.

It is apparent that lin A can be estimated from a single of the two effects A_1 or A_2. These two effects, however, do not provide the same information. An estimate $\hat{e}(A_1)$ with a variance σ^2/n (where n is the size of the fraction) produces an estimate $\sqrt{5}\hat{e}(A_1)/2$ of

name	:	QUANT1
nb. of units	:	64
Selection of basic factors	:	standard
(which define the unit)		
factor decomposition type	:	maximum
...		

Added factors		
fac.	lev. nb.	block
A	4	
B	4	
C	2	
D	2	
E	2	
F	2	
BL	4	←

Model parts
P: A_1+A_2+B_1+B_2+C+D+E+F
Models
P.P+A+B+BL
Part to be estimated
P.P+A+B+BL

Design definition

key matrix

	A_1	A_2	B_1	B_2	C	D	E	F	BL_1	BL_2
2_1	1	0	0	0	1	0	0	1	1	0
2_2	0	1	0	0	1	0	0	1	0	1
2_3	0	0	1	0	1	0	0	0	1	0
2_4	0	0	0	1	1	0	0	0	0	1
2_5	0	0	0	0	0	1	0	1	1	0
2_6	0	0	0	0	0	0	1	1	0	1

defining relations

C	=	$A_1A_2B_1B_2$
F	=	A_1A_2DE
BL_1	=	A_1B_1D
BL_2	=	A_2B_2E

Table 34: 1/4 fraction of a $4 \times 4 \times 2^4$, fitted to 4- levels quantitative factors

$$\begin{array}{lcllcl} A_1 & = & 2\lin A/\sqrt{5} & B_1 & = & 2\lin B/\sqrt{5} \\ A_1A_2 & = & \mathrm{quad}\, A & B_1B_2 & = & \mathrm{quad}\, B \\ A_2 & = & \lin A/\sqrt{5} & B_2 & = & \lin B/\sqrt{5} \end{array}$$

$$\begin{array}{lcl} A_1B_1 & = & 4\lin A \lin B/5 \\ A_1B_2 & = & 2\lin A \lin B/5 \\ A_2B_1 & = & 2\lin A \lin B/5 \\ A_2B_2 & = & \lin A \lin B/5 \end{array}$$

Table 35: Expression of pseudofactorial effects in a degree 2 model

$\lin A$ with a variance $(5/4)\sigma^2/n$, while an estimation $\hat{e}(A_2)$ with the same variance σ^2/n produces an estimate $\sqrt{5}\hat{e}(A_2)$ of $\lin A$ with a variance $5\sigma^2/n$ four times greater. It is thus preferable to have A_2 confounded rather than A_1 on the fraction.

One comment applies to both $\lin B$ and $\lin A \lin B$. Thus the latter effect can be estimated from each of the effects A_1B_1, A_1B_2, A_2B_1, A_2B_2. Nevertheless, the estimation variance, equal to $(25/16)\sigma^2/n$ on the basis of an estimate $\hat{e}(A_1B_1)$ with a variance σ^2/n, becomes $(25/4)\sigma^2/n$ on the basis of A_1B_2 or A_2B_1, and $25\sigma^2/n$ on the basis of A_2B_2. The information provided, by definition, inversely proportional to variance, are in the ratios 16, 4, 4, 1. When several estimates are combined, the information is combined. For example, if we have independent estimates $\hat{e}(A_1B_1)$, $\hat{e}(A_1B_2)$ with a variance σ^2/n, the combined estimate of $\lin A \lin B$

$$\frac{16}{20}\frac{5}{4}\hat{e}(A_1B_1) + \frac{4}{20}\frac{5}{2}\hat{e}(A_1B_2)$$

has a variance $(25/(16+4))\sigma^2/n$. Thus the pseudofactorial effects providing minimal information should primarily be confounded, i.e. A_2B_2 and, if necessary, A_1B_2, A_2B_1.

These considerations lead to the search for a fraction 1/2 of the full factorial design of size $16 = 4 \times 4$ which make the substantial pseudofactorial effects A_1, A_1A_2, B_1, B_1B_2, A_1B_1 confounded only with the 5 effects equal to zero. The corresponding model and part to be estimated appear under the heading *"search nb. 1"* in table 36. The model contains all the non-zero effects, that is to say in this instance, those effects displaying at most 2 pseudofactors. Unfortunately, this search fails. The request must thus be made

nom	:	QUANT2
nb. of units	:	8
Selection of basic factors	:	standard
factor decomposition type	:	maximum
Backtrack search - time limit	:	10 mn
- max. nb. sol.	:	999
...		

Added factors	
fac.	nb. niv.
A	4
B	4

Search nb. 1

Model parts
P:A_1+A_2+B_1+B_2
Model
P.P
Part to be estimated
A_1+A_1.A_2+B_1+B_1.B_2+A_1.B_1
FAILURE OF THE SEARCH Search stopped on factor B_2

Search nb. 2

Model
A_1.A_2+B_1.B_2+A_1.B_1
Part to be estimated
A_1+A_1.A_2+B_1+B_1.B_2+A_1.B_1
Defining relation
$A_2B_2 = 1$

Table 36: 1/2 fraction of a 4×4, with quantitative 4-level factors

more flexible by authorizing confounding of the 5 above-mentioned substantial effects with minor effects such as A_2B_1, A_1B_2, A_2B_2. The corresponding model and part to be estimated are specified under the title *"search nb 2"* in table 36. The study of aliases for

the solutions obtained shows that there is in fact only one solution defined by the relation $A_2B_2 = 1$. For this solution, the basic estimable linear combinations are

$$\begin{array}{lcll}
E\left(\langle y, \mathbf{1}\rangle/8\right) & = & e(\mathbf{1}) & + \; e(A_2.B_2) \\
E\left(\langle y, A_1\rangle/8\right) & = & e(A_1) & + \; e(A_1.A_2.B_2) \\
E\left(\langle y, A_2\rangle/8\right) & = & e(A_2) & + \; e(B_2) \\
E\left(\langle y, A_1.A_2\rangle/8\right) & = & e(A_1.A_2) & + \; e(A_1.B_2) \\
E\left(\langle y, B_1\rangle/8\right) & = & e(B_1) & + \; e(A_2.B_1.B_2) \\
E\left(\langle y, A_1.B_1\rangle/8\right) & = & e(A_1.B_1) & + \; e(A_1.A_2.B_1.B_2) \\
E\left(\langle y, A_2.B_1\rangle/8\right) & = & e(A_2.B_1) & + \; e(B_1.B_2) \\
E\left(\langle y, A_1.A_2.B_1\rangle/8\right) & = & e(A_1.A_2.B_1) & + \; e(A_1.B_1.B_2)\,,
\end{array} \tag{7}$$

where y is the vector of the 8 observations and $\langle\ \rangle$ denotes the usual scalar product of $\mathbb{R}^8$. The vectors $\mathbf{1}$, A_1, ..., A_1A_2, ... appearing in the scalar products are the vectors of 1 and -1 naturally associated in multiplicative notation with the corresponding pseudofactors or products of pseudofactors.

In view of the equalities in table 35 and since pseudofactorial effects including 3 or 4 symbols are zero, the system (7) is rewritten in the form

$$\begin{array}{lcll}
E\left(\langle y, \mathbf{1}\rangle/8\right) & = & e(\mathbf{1}) & + \; \lin A \lin B/5 \\
E\left(\langle y, A_1\rangle/8\right) & = & 2\lin A/\sqrt{5} & \\
E\left(\langle y, A_2\rangle/8\right) & = & \lin A/\sqrt{5} & + \; \lin B/\sqrt{5} \\
E\left(\langle y, A_1.A_2\rangle/8\right) & = & \operatorname{quad} A & + \; 2\lin A \lin B/5 \\
E\left(\langle y, B_1\rangle/8\right) & = & 2\lin B/\sqrt{5} & \\
E\left(\langle y, A_1.B_1\rangle/8\right) & = & 4\lin A \lin B/5 & \\
E\left(\langle y, A_2.B_1\rangle/8\right) & = & 2\lin A \lin B/5 & + \; \operatorname{quad} B \\
E\left(\langle y, A_1.A_2.B_1\rangle/8\right) & = & 0 &
\end{array} \tag{8}$$

The scalar products between y and A_1, B_1, $A_1.B_1$ enable direct estimation of the linear effects $\lin A$ and $\lin B$ and the interaction $\lin A \lin B$. The estimates of the general mean and of the two quadratic effects are inferred from these estimates and of the scalar products with $\mathbf{1}$, $A_1.A_2$, $A_2.B_1$.

Table 37 provides the estimates thus obtained, together with their associated variances and efficiencies. The calculation of variances is based on the fact that the scalar products on the left-hand side of (8), $\langle y, \mathbf{1}\rangle/8$, $\langle y, A_1\rangle/8$, $\langle y, A_2\rangle/8$, ... are uncorrelated, with a variance $\sigma^2/8$. The comparison of variances thus obtained with variance $\sigma^2/16$ of the polynomial effects in the full factorial design provides - after adjustment for the fact that there are half as many observations in the fraction - the efficiencies appearing on the right-hand side of table 37.

The notation $e()$ is used to represent polynomial effects in this table. Thus the linear effect $\lin A$, the interaction $\lin A \lin B$ are represented in the table by $e(\lin A)$, $e(\lin A \lin B)$. This notation, adopted in [7] to distinguish polynomial pseudofactors from their effects, is also convenient to represent the estimates by placing a circumflex or tilde over the e.

The information provided by $\langle y, A_2\rangle/8$ on $\lin A$ and $\lin B$ has not been used. It may be used to obtain the least square estimators of these two parameters. For the purpose

	estimation	variance	factor efficiency
$\hat{e}(\lin A)$	$= (\sqrt{5}/2)\,\langle y, A_1\rangle\,/8$	$(5/4)\,\sigma^2/8$	$4/5$
$\hat{e}(\lin B)$	$= (\sqrt{5}/2)\,\langle y, B_1\rangle\,/8$	$(5/4)\,\sigma^2/8$	$4/5$
$\hat{e}(\lin A \lin B)$	$= (5/4)\,\langle y, A_1B_1\rangle\,/8$	$(25/16)\,\sigma^2/8$	$16/25$
$\hat{e}(\mathbf{1})$	$= \Big(\langle y, \mathbf{1}\rangle - (1/4)\,\langle y, A_1B_1\rangle\Big)/8$	$(17/16)\,\sigma^2/8$	$16/17$
$\hat{e}(\mathrm{quad}\, A)$	$= \Big(\langle y, A_1A_2\rangle - (1/2)\,\langle y, A_1B_1\rangle\Big)/8$	$(5/4)\,\sigma^2/8$	$4/5$
$\hat{e}(\mathrm{quad}\, B)$	$= \Big(\langle y, A_2B_1\rangle - (1/2)\,\langle y, A_1B_1\rangle\Big)/8$	$(5/4)\,\sigma^2/8$	$4/5$

Table 37: Estimates of polynomial effects in the fraction $4^2/2$

of the calculation, only the lines associated with the scalar products $\langle y, A_1\rangle\,/8$, $\langle y, A_2\rangle\,/8$, $\langle y, B_1\rangle\,/8$ need to be taken into account in the system (8). They are rewritten in the following matrix form

$$E\begin{bmatrix} \langle y, A_1\rangle\,/8 \\ \langle y, A_2\rangle\,/8 \\ \langle y, B_1\rangle\,/8 \end{bmatrix} = \frac{1}{\sqrt{5}}\begin{bmatrix} 2 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix}\begin{bmatrix} \lin A \\ \lin B \end{bmatrix},$$

and lead to the estimates provided in table 38.

	estimation	variance	factor efficiency
$\tilde{e}(\lin A)$	$= (\sqrt{5}/12)\Big(5\,\langle y, A_1\rangle + 2\,\langle y, A_2\rangle - \langle y, B_1\rangle\Big)/8$	$(25/24)\sigma^2/8$	$24/25$
$\tilde{e}(\lin B)$	$= (\sqrt{5}/12)\Big(-\langle y, A_1\rangle + 2\,\langle y, A_2\rangle + 5\,\langle y, B_1\rangle\Big)/8$	$(25/24)\sigma^2/8$	$24/25$

Table 38: Least square estimates of the linear effects in the fraction $4^2/2$

Instead of improving the estimation of $\lin A$ and $\lin B$ with the information provided by $\langle y, A_2\rangle\,/8$, the pseudofactor A_2 can be used to divide the design into two blocks. If the block factor is denoted by C, we obtain $C = A_2$ and the row associated with A_2 in the system (8) becomes

$$E\left(\langle y, A_2\rangle\,/8\right) = \lin A/\sqrt{5} + \lin B/\sqrt{5} + e(C)\,.$$

The information provided by $\langle y, A_2\rangle\,/8$ can thus be used to estimate the block effect $e(C)$:

$$\hat{e}(C) = \langle y, A_2\rangle\,/8 - \Big(\hat{e}(\lin A) + \hat{e}(\lin B)\Big)/\sqrt{5}$$

This distribution into two blocks is useful if the loss of information on the linear effects which is induced is offset by the reduction in the residual variance σ^2.

The pseudofactor $A_1A_2B_1$ can also be used to define a second two-block system crossed with the former system. The drawback, then, is that there are no degrees of freedom left to estimate the error.

The practical relevance of this small example is limited, but it effectively shows the flexibility of this construction method. The example in the following paragraph, taken from [7], illustrates the use of this method to create a design that is considerably smaller in size.

4.2.3.2 1/16 **fraction of a** $4^3 2^4$ **with two quantitative factors** With a view to optimizing the culture medium of a rhizobial symbiont of soya (*Bradyrhizobium japonicum*), 7 composition factors from this medium are studied, including three 4-level factors and 4 2-level factors. Out of the three 4-level factors, A is qualitative, and the two others, B and C are quantitative. Table 39 specifies the search performed.

name	:	RHIZO6
nb. of units	:	64
Selection of basic factors (which define the unit)	:	user
factor decomposition type (en pseudofacteurs)	:	maximum
Backtrack search – time limit	:	20 mn
- max. nb. sol.	:	999
- random link	:	0
Inclusion of factors in the ineligible set	:	yes

Basic factors	
fac.	nb. niv.
A	4
B	4
C	4

Added factors	
fac.	nb. niv.
D	2
E	2
F	2
G	2

Model parts	
1	p: A+B_1+C_1+D+E+F+G
2	q: A+B+C+D+E+F+G

Models	
1	p.p
2	q.q

Part to be estimated	
1	p.p
2	q

Table 39: 1/16 fraction of a $4^3 2^4$ including 2 quant. 4-lev. fact.

The first couple *model, part to be estimated* ensures that the main effects and major interactions between two factors are not confounded among themselves. All the interactions which exhibit neither of the two pseudofactors B_2 and C_2 are considered to be major.

The second couple *model, part to be estimated*, in addition, requires that the design be of resolution 4, that is to say, which enables the estimation of all the main effects in a model containing all the two-factor interactions. The quantitative aspect is not taken into account in this second couple, introduced to ensure maximum robustness in estimating main effects of all degrees.

The full, exhaustive search, requested by giving 999 as the maximum number of

solutions, generates a set of 1152 solutions. The analysis of these solutions, conducted by a specific program, showed that they were all derived from 6 basic solutions by switching factors or pseudofactors with symmetric roles, that is to say, D, E, F, G first, then B and C, and finally, A_1, A_2, A_1A_2. These 6 basic solutions are provided in table 40, where three overall efficiency measurements are also reported, so that these solutions may be compared with regard to variance.

Table 41 specifies the efficiencies, for each effect, of the best three designs for overall efficiency. In the third design, interactions including A, each of which have three degrees of freedom, are characterized by three efficiencies, referred to as main efficiencies. The specific definition of these efficiencies is given in [17] and [18]. The lowest of these 3 efficiencies is the lower bound of efficiencies for all the contrasts belonging to this interaction.

plan	defining relations	trace	det	min. eigenval.
1	$A_1B_1B_2C_1D = 1$; $A_2B_1B_2C_2E = 1$ $A_2B_1C_1C_2F = 1$; $A_1B_2C_1C_2G = 1$	0.948	0.976	0.434
2	$A_1B_1B_2C_1D = 1$; $A_2B_1B_2C_2E = 1$ $A_1B_1C_1C_2F = 1$; $A_1A_2B_2C_1C_2G = 1$	0.934	0.969	0.460
3	$A_1B_2C_1D = 1$; $A_2B_2C_2E = 1$ $A_1B_1C_1C_2F = 1$; $A_1A_2B_1B_2C_1C_2G = 1$	0.912	0.958	0.460
4	$A_1B_2C_1D = 1$; $A_2B_2C_2E = 1$ $A_1B_1C_1C_2F = 1$; $A_1A_2B_2C_1C_2G = 1$	0.900	0.956	0.330
5	$A_1B_2C_1D = 1$; $A_2B_1C_2E = 1$ $A_1B_1B_2C_2F = 1$; $A_2B_2C_1C_2G = 1$	0.883	0.946	0.400
6	$A_1B_2C_1D = 1$; $A_1B_1C_2E = 1$ $A_2B_1B_2C_2F = 1$; $A_1A_2B_2C_1C_2G = 1$	0.874	0.944	0.330

Table 40: Global factor efficiencies for the 6 solutions

Effet	Factor efficiency		
	Design 1	Design 2	Design 3
A	1	1	1
$\text{lin}\, B, \text{lin}\, C$	1	1	1
$\text{quad}\, B, \text{quad}\, C$	1	1	1
D, E, F, G	1	1	1
$A.\text{lin}\, B$	1	1	1 0.840 0.800
$A.\text{lin}\, C$	1	1	1 0.960 0.840
AD	1	1	1 1 0.833
AE	1	1	1 1 0.952
AF, AG	1	1	1 1 0.800
$\text{lin}\, B.\text{lin}\, C$	0.640	0.960	0.800
$\text{lin}\, B.D$	0.840	0.800	0.840
$\text{lin}\, B.G$	0.840	0.800	0.800
$\text{lin}\, B.E$	0.840	0.800	0.960
$\text{lin}\, B.F$	0.840	0.960	0.800
$\text{lin}\, C.D$	0.840	0.960	0.840
$\text{lin}\, C.E, \text{lin}\, C.F, \text{lin}\, C.G$	0.840	0.800	0.800
DE	1	0.800	1
DF	1	0.960	1
DG	0.810	0.800	1
EG	1	0.667	0.800
EF	0.810	0.800	0.800
FG	1	0.800	0.667

Table 41: Factor efficiencies in the 3 optimal solutions for the trace

5 Use of the program

5.1 System requirement

`PLANOR` is a R package that includes routines written in C. It runs under Linux or Windows systems and requires the `conf.design`, `bigmemory` and `biganalytics` packages.

5.2 Help

The user can have access to online assistance by using the help function. The distribution includes a reference manual and a vignette, in addition to the present document.

The design search and some summary functions can be very long to execute. So we advise the user to start with small examples and to ensure he or she knows how to stop the execution of a R function in his or her R environment. At present, no result is saved when a `PLANOR` function is stopped, but information is given on how far the design search succeeded.

5.3 Typical usage

This subsection describes shortly different possible options when using the `PLANOR` package. More details are given in the next subsections on

- the creation or modification of a regular design (subsection 5.4);
- randomization (subsection 5.5);
- the study of aliases (subsection 5.6).

During the *creation of a regular design*, the user introduces the parameters defining the search in three steps:

- parameters related to the factors (function `planor.factors`);
- parameters related to the model(s) (function `planor.model`);
- parameters related to the experimental units (function `planor.designkey`).

All the information may also be provided directly to the latter function, which starts the design search. The parameters are stored in R objects. The results of the search, once it is completed, are stored in a list of key matrices. All these R objects belong to classes which are specific to `PLANOR` .

By default of the argument `max.sol` of `planor.designkey`), only a single key matrix solution is required. The user may then proceed directly to the design creation by using

the function `planor.design`. The design can be randomized at this stage, or later by using the function `planor.randomize`.

Whether one or several matrices are sought, the solutions obtained may be examined by the *study of aliases*. This can be done by applying the `summary` function to the design key objects. The systematic or randomized designs corresponding to the solution adopted are then constructed by using `planor.design`.

Several subsidiary operations may prove to be necessary in order to develop – from one or several regular designs - the appropriate design for a particular situation. The module *Recoding, selection of factors, sorting, ...* of the `PLANOR` original version made it possible to perform several of these operations simply: creation of a new factor produced from pseudofactors, recoding of levels, elimination of useless factors, repetition of certain units, merging of certain designs, sorting. These possibilities are not implemented in the `PLANOR` R package, but most of them can be managed by using the basic R functions.

5.4 Creation or modification of a regular design

5.4.1 Main steps

See Section 2.1.2.

5.4.2 Selection of basic factors

The `base` argument of the function `planor.designkey` allows to define basic factors, whose combinations of levels serve to identify the different experimental units. These basic factors can be *pseudofactors* used only to identify the units, but which have no real physical sense and do not appear in any model. This is the default option, in which case the program introduces a pseudofactor for each prime appearing in the decomposition of the number of units into primes. The program gives these pseudofactors the standard `*U*` name.

This *default* option provides considerable flexibility since it does not predetermine any of the factors (except the first one, which is forced to be identified to the first unit factor). Instead, it determines them freely as a combination of the basic factors. However, for the same reason, this is also the option which leads to the longest searches. Therefore, in many cases, the preferred option should consist in a selection of basic factors among the really active factors corresponding to treatments or blocks and appearing in the model or hierarchies. The basic factors are then entered using an additive formula such as `base= A+B+C`.

By definition, all the combinations of levels of basic factors appear in the design, as a result of which there cannot be any defining relation in which only these factors appear. The selection of basic factors by the user thus excludes certain defining relations and is likely to restrict the overall number of achievable solutions.

Consequently, when this option is selected, efforts are made from the outset to choose

basic factors for which all the combinations of levels are sought to appear in the design. The fact that these factors are chosen as basic factors does not therefore introduce any additional constraint.

For instance, if the units are structured by block systems (blocks, sub-blocks, rows, columns, etc ...), the corresponding block factors may be introduced as basic factors, potentially completed by a pseudofactor associated with the repetition number. Similarly, in the search for a distribution into blocks of a full factorial design, the treatment factors may be taken as basic factors, potentially completed by a pseudofactor associated with the repetition number if the treatments are repeated.

In all cases, if only one solution is required, any choice of basic factors for which the search completes is acceptable. It is only when the search fails that the question should sometimes be raised as to whether such failure is not the result of a flawed selection of basic factors.

5.4.3 Factor decomposition

The factors which do not have a prime number of levels are decomposed into products of several pseudofactors. In the `PLANOR` R package, this decomposition leads to pseudofactors whose numbers of levels are all prime. This is more restrictive than in the original `PLANOR` version, which could also handle decompositions into prime powers with powers larger than one. More precisely, the `PLANOR` R package behaviour corresponds to the *maximum* option of the factor decomposition type defined in the original `PLANOR` version.

For example, Table 42 provides the possible decompositions of a 24-level factor A in the original `PLANOR` version. The most comprehensive decomposition was obtained with

maximum	minimum	free choice
A_1 (2)	A_1 (8)	A_1 (4)
A_2 (2)	A_2 (3)	A_2 (2)
A_3 (2)		A_3 (3)
A_4 (3)		

Table 42: Pseudofactor decomposition of a 24-level factor A

option *maximum*, the standard option then proposed by the program. All the pseudofactors obtained in this decomposition have a prime number of levels. On the contrary, option *minimum* provided the most restricted number of pseudofactors. The numbers of pseudofactor levels obtained in this option were the highest prime powers dividing the number of levels of the decomposed factor. Option *free choice* allowed to choose the decomposition used for each factor, freely and independently.

From a practical point of view, the *maximum* option retained in the `PLANOR` R package provides the greatest flexibility. This can be observed empirically and it has been demonstrated for certain classes of full factorial designs distributed into blocks [29], [30].

5.4.4 Backtrack search

To explain the meaning of the two arguments `randomsearch` and `max.sol`, we concisely describe this search in a case when the numbers of levels are all powers of the same prime. A detailed description of the general case is presented in [19]. Note that the *time limit* parameter of the original `PLANOR` version has not (yet?) been implemented in the R package.

The program initially establishes the lists $\mathcal{L}_1$, $\mathcal{L}_2$, ..., $\mathcal{L}_s$ of possible vectors for each of the columns 1, 2, ... s of the key matrix.

It then successively searches the columns of the key matrix so as to respect the constraints imposed. Once columns 1 to i which are compatible with the constraints have been selected, the permissible vectors are then inferred from list $\mathcal{L}_{i+1}$. If this set of vectors is non-empty, the program selects the first vector as column $i+1$ and continues in a sequence. If this set is empty, it selects the next permissible element, if there is one, from list $\mathcal{L}_i$. If there is none, it reverts to the column $i-1$ choice, and so forth.

When there is no solution, the procedure terminates when list $\mathcal{L}_1$ is exhausted, which means that all the possible choices have been examined. As this exhaustive analysis may, in some cases, take a considerable time to achieve, a *time limit* was provided in the initial `PLANOR` version beyond which the program terminated. The user then had the possibility of either continuing the search by increasing the time limit or of effectively stopping the search. This is not implemented in the `PLANOR` R package.

It is clear that the order in which the lists $\mathcal{L}_i$ are classified have a determining influence on the choice of the key matrix in cases in which only one solution is sought. To avoid obtaining the same solution systematically, it is possible to reorder these lists randomly by specifying `randomsearch=TRUE` in the function `planor.designkey`. This generates random reordering of these lists and thus the solution obtained is random too. Note that the same search performed with the same seed value of the random number generator of R always leads to the same solution (see the help of function `set.seed` for more information).

In many cases, it is appropriate to search the best possible solution among all solutions satisfying the constraints, with respect to a criterion not taken into account in the search. For instance, among all the fractions of resolution 4, that which has the smallest possible number of interactions confounded may be sought. In this case, it would be appropriate to first obtain all the possible solutions. For this purpose, a comprehensive search can be initiated by specifying `max.sol=Inf` in the function `planor.designkey`. In order to find all the solutions, the program then continues the search after obtaining a solution, by moving to the next permissible element in list $\mathcal{L}_s$ associated with the last column s of the key matrix. If there is none, it returns to column $s-1$ and selects the next permissible element, and so forth. It terminates either when list $\mathcal{L}_1$ is exhausted (or after the time limit was reached in the initial version).

The disadvantage of this comprehensive search procedure is that it can take a very long time to achieve (particularly since, in this version of the program, symmetries between factors are not taken into account to reduce the search time). Moreover, the solutions

obtained are often very close and quite often differ only by one of the columns of the key matrix. Premature termination of the searches thus leads to an insufficiently diversified set of solutions.

In order to obtain a range of markedly different solutions without performing the comprehensive search, it is sufficient to request a reduced number of solutions (strictly below infinity in any case, which can be specified by `max.sol=Inf`) and a random search. In this case, the search procedure is restarted after each solution, with random reordering of the lists $\mathcal{L}_i$[4]. The solutions obtained are stored in the output of function `planor.designkey`, a solution can be selected by its number to explicitly construct the design.

The backtrack search thus uses the following two arguments :

- `randomsearch` It determines the order in which possible solutions are explored for each column of the key matrix : lexicographic order if `randomsearch=TRUE` and random order if `randomsearch=TRUE`.
- `max.sol` If it is equal to `Inf`, the backtrack search continues until all the solutions are obtained (or until the time limit was reached). Such a comprehensive search has a chance of being completed only for small-scale problems.

There is presently no time limit parameter in the R `PLANOR` package. When no solution actually exists, continuation of the search can take a considerable length of time because it is necessary to explore all the possibilities in order to prove the absence of a solution. In practice, it is therefore not advised to continue a search for too long.

5.4.5 Inclusion of factors in the ineligible set

The data for the model and part to be estimated makes certain defining relations *ineligible*, that is to say, prevents these relations from being used to define the design. Nevertheless, if the main effect of a factor does not appear in the part to be estimated, the search may produce a defining relation in which only this single factor appears, in which case only a fraction of the levels then appears in the design. In certain cases this is not acceptable, for instance if it is a block factor in which all the levels must be present.

Forcing the systematic inclusion of factors in the ineligible set was optional in the original `PLANOR` version but it is imposed in the `PLANOR` R package. This makes any defining relation involving only a single factor *ineligible* and thus ensures that each factor assumes all its levels in the design.

5.4.6 Factors and models

5.4.6.1 Basic factors and added factors It is possible to indicate that a factor is a block factor by using the `block` argument of the function `planor.designkey`. This

[4]actually this is not the present behaviour in the R `PLANOR` package because there is presently no random reordering after each solution

indication is not used in the backtrack search. It is only used to distinguish between block effects and treatment effects in the study of aliases. In the initial `PLANOR` , it was also used in the randomization phase, but in the `PLANOR` R package the factorial structure to be used for randomization has to be defined explicitly and it can be independent of the block factor declarations.

The levels of the defined factors are deduced from those of the basic factors by using the rules defined by the key matrix. This key matrix is sought through the backtrack procedure so as to make the parts to be estimated in the associated models estimable and to respect the potential hierarchies. Some of these factors can be predetermined by the user. The associated columns of the key matrix are then predetermined in the search (not yet implemented).

The remainder of this paragraph briefly recalls the functions and syntaxes associated with the arguments `model`, `resolution` and `hierarchy` of the function `planor.designkey`. A more pedagogical introduction to these notions through examples is featured in Section 2 and more general descriptions may be found in [18], [19].

5.4.6.2 Models and parts to be estimated Usually, a single `model` providing the non-negligible factorial effects is introduced. The "*part to be estimated*" of this model is then specified in the `estimate` argument, if different from the model. This is done using the R syntax for writing model formulae. Alternatively, the `resolution` argument can be used as a short-hand declaration when the problem is symmetric with respect to the factors.

It may be necessary to introduce several models and corresponding parts to be estimated, to deal with cases in which block systems induce several strata for instance.

It is also possible to indicate *hierarchy* constraints among factors by using the `hierarchy` argument.

A model in `PLANOR` must be complete with respect to marginality between factorial effects. Thus if the model includes the interaction $A.B$, then it must also include the main effects A, B and the general mean. This is not necessarily true for the "part to be estimated". If it contains the term $A.B$ but not the terms A and B, this indicates that the interaction $A.B$ is sought to be estimated, but not necessarily the main effects A and B.

When the `estimate` argument is restricted to the general mean (`estimate=~1`), it follows that the terms of the associated completed model cannot be confounded with the general mean, which precludes any defining relation based only on the factors of one of these terms. These defining relations are *ineligible*. The use of a part to be estimated limited to the general mean particularly ensures that the products of factors that appear in the associated model take all their levels in the design.

Similarly, when the `model` argument is restricted to the general mean (`model=~1`), the terms of the associated part to be estimated cannot be confounded with the general mean and result in ineligible defining relations. Since the part to be estimated may be incomplete with respect to marginality, this use of `estimate=~1` provides greater flexibility

in introducing ineligible defining relations than the use of `model=~1` .

5.4.6.3 Hierarchies If the level of a factor A must remain constant for each combination of levels of certain other factors, say B, C, D, this is considered as a hierarchical constraint between factors. Such a constraint can be specified by the argument `hierarchy =~A/(B*C*D)` in the function `planor.designkey`. It is then said that the product factor $B \times C \times D$ is nested within A or, alternately, that A is marginal to this factor.

Example. In a design containing blocks –BL– and sub-blocks –SBL–, the factor VAR is constant for each block and the factor DOSE is constant for each sub-block. This double constraint can be indicated by

```
hierarchy=list(~VAR/BL, ~DOSE/(BL*SBL)
```

where SBL provides the number of the sub-block within the block.

5.4.6.4 Example with several models and hierarchies In a block design in which certain factors cannot vary within blocks, it is often indispensable to introduce several models and parts to be estimated as illustrated in the following, typical example :

```
> typical <- planor.designkey(
+   factors=c(LETTERS[1:8], "BL"),
+   nlevels=c(rep(2,8),8),
+   hierarchy=list(~A/BL, ~B/BL, ~C/BL, ~D/BL),
+   listofmodels=list(
+     noblock.model=c(~(A+B+C+D)^2, ~A+B+C+D),
+     block.model=c(~BL + (A+B+C+D+E+F+G+H)^2,
+                   ~(A+B+C+D+E+F+G+H) : (E+F+G+H))),
+   nunits=64, base=~BL+E+F+G)


Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  => search for columns 7 to 11
      first visit to column 7
      first visit to column 8
      first visit to column 9
      first visit to column 10
      first visit to column 11
The search is closed: max.sol =  1 solution(s) found


> print(typical)
```

```
An object of class listofkeyrings

********** Prime  2  design **********

--- Solution  1  for prime  2  ---

      E F G BL_1 BL_2 BL_3 A B C D H
E     1 0 0    0    0    0 0 0 0 0 1
F     0 1 0    0    0    0 0 0 0 0 1
G     0 0 1    0    0    0 0 0 0 0 1
BL_1  0 0 0    1    0    0 1 0 0 1 1
BL_2  0 0 0    0    1    0 0 1 0 1 0
BL_3  0 0 0    0    0    1 0 0 1 1 0
```

The part to be estimated $(E + F + G + H)$ in the so-called "block model" includes the main effects of E, F, G, H and their interactions with each of the 8 treatment factors A, B, C, D, E, F, G, H (i.e. 26 terms in developed form).

The main effects A, B, C, D cannot be estimated in the model including the block effect (BL) since they cannot vary within blocks. The so-called "noblock model" nevertheless ensures their estimability in an inter-block model including all the interactions between two of these factors.

5.4.6.5 How to force a product of factors to assume all its levels As has been indicated, a product of factors can be forced to assume all its levels in the design by making it appear as the term of a model associated with an estimate part reduced to the general mean.

As an example let us consider a situation in which there are three treatment factors A, B, C with 6, 6 and 2 levels. We further assume that the 72 treatments of the full factorial design must be distributed within a cube containing 6 positions on the X-axis, 3 on the Y-axis and 2 heights –factors X, Y, Z–. To illustrate this, we may imagine we are cultivating 6 strains of *champignons de Paris* (button mushrooms) (factor A) with 6 different composts (B) with 2 pH values (C)and that these mushrooms are distributed in the growth chamber over two overlapping trays (Z), each containing 6 rows (X), 3 columns (Y) with two containers in each location defined by X, Y, Z.

Taking into consideration the ventilation system and the position of the growth chamber door, it is estimated that the three factors X, Y, Z are likely to have an effect. We wish to be able to estimate the main effects A, B, C and interactions $A.C$, $B.C$ in a model containing the additive effects X, Y, Z and all the interactions between treatment factors.

If the basic factors used are X, Y, Z and U is the container number in the pair, we can be sure that each of the 36 triplets of coordinates X, Y, Z appears twice. However, if we use A, B, C as the basic factors, the following pair `model, estimate`:

model	part to be estimated
[1] $X + Y + Z + A.B.C$	[1] $A + B + C + A.C + B.C$

may result in defining relations in which all the triplets of coordinates X, Y, Z do not appear, such as those in table 43.

2-lev. factor			
	X_1	Z	
A_1	1	1	$X_1 = A_1B_1C$
B_1	1	1	$Z = A_1B_1$
C	1	0	
3-lev. factor			
	X_2	Y	$X_2 = A_2^2B_2^2$
A_2	2	2	$Y = A_2^2B_2^2$
B_2	2	2	

Table 43: An inappropriate solution to dispatch in 3 crossed block systems

In order to avoid such inappropriate relations, we introduce a model including the term $X.Y.Z$ associated with a blank (empty) "part to be estimated":

$$\begin{array}{cc} \text{model} & \text{part to be estimated} \\ [2] \quad X.Y.Z & [2] \quad 1 \text{ (general mean)} \end{array} \tag{9}$$

The choice of A, B, C as basic factors is quite natural. Firstly, it does not make the introduction of an additional pseudofactor such as U compulsory. Secondly, this is the option which is spontaneously used when seeking to find out, independently of any program, the possible constructions. The question is then raised as to which interactions between treatment factors each of the three block systems should be confused with, which results in the block pseudofactors being defined from the treatments rather than the reverse.

5.4.6.6 Predetermined factors (not yet implemented in the PLANOR R package)

The following lines:

```
Bl_1 : A_1 + B_1 + C_1
Bl_2 : A_2 + 2B_2
E    : 2A_2 + D
```

in the window with the predetermined factors set the way in which the pseudofactors Bl_1, Bl_2 inferred from factor Bl and factor E are calculated from the pseudofactors A_1, B_1, C_1, A_2, B_2 inferred from the basic factors A, B, C and from factor D.

After checking that these definitions verify the constraints, the program takes them into account in the search for the other factors.

It is possible to define all the factors in this way. This makes it possible to study of aliases in a predetermined design or to construct and randomize this design.

5.5 Randomization

In agricultural experiments, which have prompted the development of experimental designs and the theory of randomization, it is generally recognized that the observation is the sum of a treatment effect and of an uncontrolled effect of the experimental unit. In order to prevent the effects of the experimental units from systematically overlapping with the effects of the treatments being compared and from skewing the comparisons, the experimental unit allocated to each treatment is chosen randomly: this is referred to as *randomization*.

In the `PLANOR` R package, randomization can be performed either by using the `randomize` argument of the function `planor.design`, or by using the function `planor.randomize` on a design. In both cases, randomization is specified by a model formula which essentially describes the block structure of the experiment.

When all the units are equivalent, randomization is totally free. This is called *complete randomization*. It consists of determining by random draw without replacement the real unit (a plot in the field of agriculture, an animal in animal science, etc ...) allocated to each unit of the systematic design. Such a randomization is obtained by using formula `~UNITS` for the randomization model. An example is provided in table 1. The numbers resulting from the draw are called repetition indices and appear in a column identified by the heading *ind-rep*.

In this example, the number of units in the systematic design is equal to the maximum number of available units and randomization equates to randomly selecting a permutation of numbers from 0 to 7, with the same probability of selecting each of the !8 permutations.

When the experimental units are structured into blocks, randomization must comply with this structure. This block structure is specified by the block factors. However, the levels given to the latter factors in the systematic design are unrelated to the labels or numbers of the real experimental units. Randomization, described below, results precisely in replacing these levels of the systematic design by the levels identifying the real experimental units. This is achieved in such a way that all the units with the same block factor level in the systematic design have the same level in the randomized design as a result of this substitution. Randomization of the ROBOT1 design in § 3.1.3 provides an example (table 10).

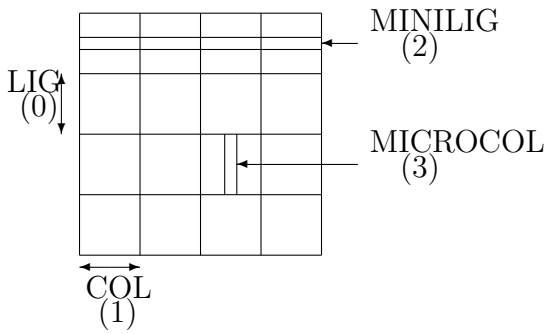
Specification of the block factors is achieved through the randomization model. These factors must have the ability to be expressed from a set of basic pseudofactors, denoted *block pseudofactors*, which are such that the number of units per combination of levels of these pseudofactors is constant. If this number of units is $k > 1$, the user can (and should, usually) introduce an additional block pseudofactor `UNIT` in the blockstructure model formula, to specify that randomization must be performed also at the unit level.

The randomization block structure is introduced as a model formula. A random seed can be introduced prior to the randomization by the base R command `set.seed`. In that case, the randomization performed is entirely determined by the random seed, which

enables the same randomization to be performed again, identically: this only requires keeping the same random seed. It should be noted that if the randomized file is lost, it can only be identically recreated if the corresponding random seed is known.

In the example in 5.5, the plate factor coincides with pseudofactor *pl*. Nevertheless, pseudofactor *col* has no significance in this example if it is separated from factor *pl*. The columns are considered here as a subdivision of the plates. There is a total of 6×4 columns. Each column is specified by the number of the plate to which it belongs *pl* and its number *col* in this plate. The randomization model thus comprises two terms, *pl* and *pl.col*, with the second term corresponding to the column factor. Randomization is described for this particular example in paragraph 3.1.3.

Another example in a similar situation would include the following four block factors: rows (LIG) crossed with columns (COL), mini-rows nested within rows (MINILIG) and micro-columns (MICROCOL) subdividing the cells formed by intersection of the rows and columns (figure 3). In this example, the units are defined by the quadruplet of levels of



Randomization model

LIG+COL+MINILIG.LIG+MICROCOL.LIG.COL

Partial order between pseudofactors

MINILIG < LIG	{(2)< (0)}
MICROCOL < LIG	{(3)< (0)}
MICROCOL < COL	{(3)< (1)}

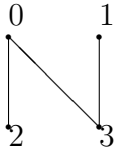


Figure 3: Example of block structure (extracted from [4])

the 4 pseudofactors LIG, COL, MINILIG, MICROCOL. The factors row, column coincide with the pseudofactors LIG, COL respectively. The factors mini-row and micro-column are defined by the products MINILIG.LIG and MICROCOL.LIG.COL, respectively.

In the general case, the model is thus formed by the list of block factors which are

defined as products of the block pseudofactors.

```
> set.seed(123)
> bprs.key <- planor.designkey(factors=list(LIG=c("L1","L2","L3"),
+                                  MINILIG=1:2,
+                                  COL=c("C1","C2"),
+                                  MICROCOL=1:2),
+                                model=~LIG*COL*MINILIG*COL*MICROCOL,
+                                estimate=~1,
+                                nunits=24,
+                                base=~LIG+MINILIG+COL+MICROCOL)

Preliminary step 1 : processing the model specifications
Preliminary step 2 : performing prime decompositions on the factors
Main step for prime p = 2 : key-matrix search
  no need (all columns are predefined)
Main step for prime p = 3 : key-matrix search
  no need (all columns are predefined)

> unrand <- planor.design(bprs.key)@design
> unrand <- unrand[order(unrand$LIG,unrand$MINILIG,
+                        unrand$COL,unrand$MICROCOL),]
> unrand$U <- letters[1:24]
> cat("\n The unrandomized design\n")

 The unrandomized design

> print(t(matrix(unrand$U,4,6)))

     [,1] [,2] [,3] [,4]
[1,] "a"  "b"  "c"  "d"
[2,] "e"  "f"  "g"  "h"
[3,] "i"  "j"  "k"  "l"
[4,] "m"  "n"  "o"  "p"
[5,] "q"  "r"  "s"  "t"
[6,] "u"  "v"  "w"  "x"

> rand <- planor.randomize(
+           blockformula=~LIG+COL+LIG:MINILIG+COL:LIG:MICROCOL,
+           data=unrand)
> rand <- rand[order(rand$LIG,rand$MINILIG,rand$COL,rand$MICROCOL),]
> cat("\n The randomized design\n")

 The randomized design
```

```
> print(t(matrix(rand$U,4,6)))

     [,1] [,2] [,3] [,4]
[1,] "r"  "q"  "t"  "s"
[2,] "v"  "u"  "x"  "w"
[3,] "i"  "j"  "l"  "k"
[4,] "m"  "n"  "p"  "o"
[5,] "e"  "f"  "h"  "g"
[6,] "a"  "b"  "d"  "c"
```

When the number of units per combination of levels of pseudofactors is strictly greater than 1 ($k \geq 1$), the initial `PLANOR` used to automatically add a unit factor to the model used for randomization. In the `PLANOR` R package, the user must include explicitly a UNIT term in the randomization block formula to ensure that randomization occurs at the finest level.

From the randomization block formula, a *partial order* is immediately inferred between the block pseudofactors. By definition, $A \leq B$ is obtained if every factor of the model containing A also contains B. In particular, the repetition index, when it is introduced, is lower than all the other pseudofactors. This order is often represented in the form of a diagram, termed Hasse diagram, such as that which appears in figure 3.

In fact it is more precisely a preorder rather than an order because two distinct pseudofactors A_1 and A_2 can be systematically associated thereby giving us both $A_1 \leq A_2$ and $A_2 \leq A_1$ without the equality. However this situation, processed by the program by simply replacing each set of constantly associated pseudofactors by their product pseudofactor, may be ignored.

The structure of the block system for the example represented in figure 2, may be described by model *col1 + col1.col2 + row1 + row1.row2* to which the order *col2<col1, row2<row1* corresponds. The randomization program renumbers the pseudofactors in a manner compatible with this order. More precisely, in cases in which certain factors emerge as systematically associated in the terms of the model, first it determines the *classes* of associated pseudofactors, then it renumbers these classes. Each class is here reduced to a factor and the internal numbering used is 0 for *col1*, 1 for *row1*, 2 for *col2*, 3 for *row2*. Note that if a class comprised more than one pseudofactor, the program would replace the factors of this class by their product. It may thus always be considered in the following that there is a single pseudofactor per class.

For each pseudofactor j, the list $]j) = \{i/i > j\}$ of pseudofactors which are strictly above it is determined. In the example, lists $]0)$ and $]1)$ associated with pseudofactors 0 (*col1*) and 1 (*row1*) are empty. Lists $]2)$ and $]3)$ associated with pseudofactors 2 (*col2*) and 3 (*row2*) contain factors 0 (*col1*) and 1 (*row1*), respectively.

For each block pseudofactor j, the actual levels are determined by random drawing without replacement performed independently for each of the combinations of levels of the higher pseudofactors, that is to say, the pseudofactors of $]j)$. Thus for the column pseudofactor *col2*, a separate draw is performed for each of the two macro-columns *col1* $= 0$ and *col1* $= 1$.

Determination of the actual level of the block pseudofactors for a unit of the systematic design is based on an ad hoc draw. The latter depends, for each pseudofactor, on the levels of the higher pseudofactors. For instance, for the systematic design unit defined by *col1* = 1, *row1* = 1, *col2* = 0, *row2*= 0, determination of *col2* in the randomized design is based on the permutation of macro-column 1 since *col1* = 1. The information in table 12 reveals that the levels of the four pseudofactors in the randomized design are *col1* = 0, *row1* = 0, *col2* = 1, *row2* = 0.

The theory underlying this type of randomization is described in [4]. A simplified description appears in [3]). From the results appearing in [4], it is easily inferred that such randomization induces a covariance structure of the random effects of the units identical to that of a classic model with random effects containing a random block effect for each *ancestral* term (see [15]). Any product of pseudofactors which, when it contains a pseudofactor A, also contains all the higher pseudofactors - i.e. all the B such as $A \leq B$ - is ancestral. In the classic model in question, the effects are all uncorrelated, and their variance is constant for each effect.

To illustrate the latter point, let us consider again the different randomization models already considered in this paragraph.

- Model *pl* + *pl.col*. Each column of each plate contains 2 units and the model thus must be completed by the term *pl.col.UNIT*. The three terms of the model *pl* + *pl.col* + *pl.col.UNIT* thus completed are the only ancestral terms. The model derived from randomization thus has a covariance structure analogous to that of a model containing random effects of the plate, of the column in the plate plus an error associated with each unit. The analysis of variance of regular design normally includes a *strata* for each of these effects.

- Model LIG+COL+MINILIG.LIG+MICROCOL.LIG.COL. The ancestral terms making up the model which provides the covariance structure are the following:
  LIG+COL+COL.LIG+MINILIG.LIG+MINILIG.LIG.COL+
  +MICROCOL.LIG.COL+MICROCOL.LIG.COL.MINILIG

- Model col1+row1+col1.col2+row1.row2. The ancestral terms are:
  col1 + row1 + row1.col1 + col2.col1 + col2.col1.row1 +
  row2.row1 + row2.row1.col1 + row2.row1.col2.col1

Remark. Any addition or removal of factors in the randomization model which does not modify the partial order between pseudofactors induces the same randomization. The largest model that follows this order is that which contains all the ancestral terms. It may be obtained by forming all the possible intersections between terms, to begin with, followed by all the unions.

It should be noted that in principle the model may contain less terms than pseudofactors. Thus factor LIG may be removed from model LIG + COL + MINILIG.LIG + MICROCOL.LIG.COL. The new model, COL + MINILIG.LIG + MICROCOL.LIG.COL, induces the same partial order. In fact, the initial model is reobtained by adding the term LIG, the only term common to the last two terms and therefore equal to their intersection: LIG = {MINILIG, LIG} $\cap$ {MICROCOL, LIG, COL}.

5.6 Study of aliases

Once the key matrices responding to specifications are obtained, the aliased effects for all or part of the solutions may be found through the study of aliases by the `alias` method. The default model is that which was previously introduced for searching the design, but it is possible to specify another one.

The study of aliases differentiates the treatment factors from the block factors. It is thus important, in order to have outputs adapted to the problem posed, to indicate beforehand which are the block factors. This is done by using the argument `block` of the `planor.factors`, `planor.designkey` or `regular.design` functions.

Examples of outputs in the study of aliases abound in this manual. We will refer in particular to § 2.2.2, 2.3.2. The example of § 3.3.2 illustrates how knowledge of aliases is used to choose a good solution.

References

[1] Agrawal V., Dey A. (1983). Orthogonal Resolution-IV Designs for Some Asymmetrical Factorials. *Technometrics*, **25**, n° 2, 197-199.

[2] Ankenman B.E. (1999). Design of experiments with Two- and Four-level factors. *J. Quality Technology*, **31**, n0 4, 363-375.

[3] Bailey R.A. (1981). A Unified Approach to Design of Experiments. *J. Roy. Statist. Soc. A*, **144**, Part 2, 214-223.

[4] Bailey R.A., Praeger C.E., Rowley C.A., Speed T.P. (1983). Generalized wreath products of permutation groups. *Proc. London Math. Soc.*, (3), **47**, 69-82.

[5] Bailey R.A. (1985). Factorial Design and Abelian groups. *Linear Algebra Appl.*,**70**, 349-368.

[6] Brouillaud-Delattre A., Kobilinsky A., Cerf O. (1994). Méthode de mesure de l'efficacité des procédés de nettoyage et de désinfection des surfaces ouvertes. *Lait*, **74**, 79-88.

[7] Cliquet S., Durier C., Kobilinsky A. (1994). Principle of a fractional factorial design for qualitative and quantitative factors: application to the production of *Bradyrhizobium japonicum* in culture media. *Agronomie*, **14**, 569-587.

[8] Edmondson R.N. (1991). Agricultural response surface experiments based on four-level factorial designs. *Biometrics*, **47**, 1435-1448.

[9] Edmondson R.N. (1993). Systematic row-and-column designs balanced for low order polynomial interactions between rows and columns. *J. R. Statist. Soc. B*, **55**, 707-723.

[10] El Mossadeq A. (1993). Construction des fractions régulières de plans d'expériences factoriels. Thèse d'Etat. Université de Pau et des pays de l'Adour.

[11] Franklin M.F. and Bailey R.A. (1977). Selection of Defining Contrasts and Confounded Effects in Two-level Experiments. *Appl. Statist.*, **26**, No. 3, 321-326.

[12] Franklin M.F. (1985). Selecting Defining Contrasts and Confounded Effects in p^{n-m} Factorial Experiments. *Technometrics*, **27**, No. 2, 165-172.

[13] Fries A. and Hunter W.G. (1980). Minimum aberration 2^{k-p} designs. *Technometrics*, **22**, 601-608.

[14] Kobilinsky A. (1985) Orthogonal factorial designs for quantitative factors. *Statist. Decisions, Suppl.*, **2**, 275-285.

[15] Kobilinsky A. (1989). Randomization of a cartesian block structure. Internal document.

[16] Kobilinsky A. (2005). PLANOR : program for the automatic generation of regular experimental designs. Version 2.2 for Windows. Technical Report, MIA Unit, INRA Jouy en Josas.

[17] Kobilinsky A. (1990). Complex linear models and cyclic designs. *Linear Algebra and Applications*, **127**, 227-282.

[18] Kobilinsky A. (1997). Les plans factoriels. Chap 3, p69–209. *In: Plans d'expériences: applications à l'entreprise*, Eds : Droesbeke, Fine, Saporta. Technip, Paris. 509p.

[19] Kobilinsky A. (1994). Automatic generation of asymmetrical regular designs. Internal document.

[20] Kobilinsky A., Fliss M., Carpentier B. (1995). Méthodes et exemples de construction de plans factoriels fractionnaires. Dans: Stratégie expérimentale et procédés biotechnologiques. Collection *Récents progrès en génie des procédés*, Vol 9, **36**, p19-28, Ed: GFGP, Diffuseur: Lavoisier.

[21] Kobilinsky A. and Monod H. (1991). Experimental design generated by group morphisms: an introduction. *Scand. J. Statist.*, **18**, 119-134.

[22] Kobilinsky A. and Monod H. (1995). Juxtaposition of regular factorial designs and the complex linear model. *Scand. J. Statist* **22**, n° 2, 223-254.

[23] Margolin B.H. (1969). Resolution IV Fractional Factorial Designs. *J. Roy. Statist.Soc. Ser. B*, **31**, 514-523.

[24] McLean R.A. and Anderson V.L. (1984). Applied factorial and fractional designs. Marcel Dekker, New-York, 373p.

[25] Montgomery D.C. (1997). Design and analysis of experiments, 4th ed. John Wiley & Sons, New-York, NY.

[26] National Bureau of Standards (1957). Fractional Factorial Experiment Designs for Factors at Two Levels. Applied Mathematics Series. **48**. (Reproduced in the book by McLean and Anderson).

[27] Patterson H.D. (1976). Generation of factorial designs. *J. Roy. Statist.Soc. Ser. B*, **38**, 175-179.

[28] Patterson H.D. and Bailey R.A. (1978). Design keys for factorial experiments. *Appl. Statist.*, **27**, 335-343.

[29] Voss D.T. (1988). Single-generator generalized cyclic factorial designs as pseudofactor designs. *Ann. Statist.*, **16**, 1723-1726.

[30] Voss D.T. (1993). Comparing the classes of single replicate generalized cyclic factorial designs with and without pseudofactors. *J. Statist. Plann. Inference* **35**, 113-120.