

CONSTRUCTION OF LATTICE SQUARE DESIGNS

By

Walter T. Federer and Jill Wright

**Biometrics Unit
Cornell University
Ithaca, NY 14853**

BU-906-M

June 1986

Construction of Lattice Square Designs

By

Walter T. Federer and Jill Wright

BU-906-M*

June 1986

ABSTRACT

Khare and Federer (1981) presented a simple method for constructing incomplete block designs for any number of treatments. Their procedure is extended to constructing lattice square designs. Using variety cutting, lattice square designs are available for any number of treatments.

INTRODUCTION

Khare and Federer (1981) presented a simple construction procedure for resolvable incomplete block designs for any number of treatments. Their procedure requires no special tables or generators and in many cases produced optimal incomplete block designs with zero-one concurrences of pairs of treatments within blocks. Their procedure is extended to construct lattice square designs with k^2 treatments. An Algorithm outlining the steps in the construction is given in the next section and is illustrated with examples.

A discussion of using the procedure for nonprimes is also presented. The procedure does not always produce optimal designs for nonprimes. The variety cutting technique of Khare and Federer (1981) is used to obtain lattice square and lattice rectangle designs (See e.g. Federer, 1955, and Raktoe and Federer, 1965.) for varying numbers of treatments v .

*In the Technical Report Series of the Biometrics Unit, Cornell University, Ithaca, NY 14853.

AN ALGORITHM FOR CONSTRUCTING LATTICE SQUARE DESIGNS

The first $k+1$ steps, k =any integer, for constructing lattice square designs for $v=k^2$ treatments follow steps given by Khare and Federer (1981). The second set of $k+1$ steps produces lattice square designs.

ALGORITHM

Step 1. Write the numbers $1, 2, \dots, k^2$ consecutively in a square array of k rows and k columns beginning in the left-hand corner of the first row and continuing for numbers $1, 2, \dots, k$; then, write numbers $k+1, k+2, \dots, 2k$ consecutively in row two. Continue this process until all rows are filled in consecutive order. This forms array 1.

Step 2. Take the main right diagonal of square array 1 to form the first row of array 2. Then, complete the columns of this array by writing down the numbers of each column in cyclic order.

Step 3. Repeat the process in step 2 on array 2 to obtain array 3.

⋮

Step $k+1$: Repeat the process on the k th array to obtain array $k+1$. As a check on the procedure, repeat the process of step 2 on the $k+1$ st array produces array 2.

Step $k+2$: Take the first rows of any two of the above $k+1$ square arrays using the first row of one of the arrays to produce the first row and using the first row of the second array to produce the first column of a new square. Using the first array, complete the last $k-1$ rows by writing the numbers of array 1 in cyclic order. This forms square 1 of a lattice square design where the row grouping is that of the first array selected above and the column grouping of treatments is that of the second array obtained.

Step k+3: Transpose the rows and columns of the square obtained in step k+2 to obtain a second square of a lattice square design.

Step k+4: Repeat the process in step k+2 on two different square arrays from the first k+1 steps to produce a third square of a lattice square design.

Step k+5: Transpose the rows and columns of the previous step to obtain a fourth square of a lattice square design.

⋮

Step 2k+1: Repeat the process on the last of the $(k+1)/2$ pairs of square arrays from the first k+1 steps to obtain the kth square of a lattice square design.

Step 2k+2: Transpose the rows and columns of the square in the previous step to obtain the k+1st square of a lattice square design.

The above Algorithm is illustrated with an example for $v = 25$. The $2(5+1) = 12$ steps follow:

Steps 1-6:

Square array 1	Square array 2	Square array 3																																																																											
<table border="1"><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td></tr><tr><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr><tr><td>16</td><td>17</td><td>18</td><td>19</td><td>20</td></tr><tr><td>21</td><td>22</td><td>23</td><td>24</td><td>25</td></tr></table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	<table border="1"><tr><td>1</td><td>7</td><td>13</td><td>19</td><td>25</td></tr><tr><td>2</td><td>8</td><td>14</td><td>20</td><td>21</td></tr><tr><td>3</td><td>9</td><td>15</td><td>16</td><td>22</td></tr><tr><td>4</td><td>10</td><td>11</td><td>17</td><td>23</td></tr><tr><td>5</td><td>6</td><td>12</td><td>18</td><td>24</td></tr></table>	1	7	13	19	25	2	8	14	20	21	3	9	15	16	22	4	10	11	17	23	5	6	12	18	24	<table border="1"><tr><td>1</td><td>8</td><td>15</td><td>17</td><td>24</td></tr><tr><td>2</td><td>9</td><td>11</td><td>18</td><td>25</td></tr><tr><td>3</td><td>10</td><td>12</td><td>19</td><td>21</td></tr><tr><td>4</td><td>6</td><td>13</td><td>20</td><td>22</td></tr><tr><td>5</td><td>7</td><td>14</td><td>16</td><td>23</td></tr></table>	1	8	15	17	24	2	9	11	18	25	3	10	12	19	21	4	6	13	20	22	5	7	14	16	23
1	2	3	4	5																																																																									
6	7	8	9	10																																																																									
11	12	13	14	15																																																																									
16	17	18	19	20																																																																									
21	22	23	24	25																																																																									
1	7	13	19	25																																																																									
2	8	14	20	21																																																																									
3	9	15	16	22																																																																									
4	10	11	17	23																																																																									
5	6	12	18	24																																																																									
1	8	15	17	24																																																																									
2	9	11	18	25																																																																									
3	10	12	19	21																																																																									
4	6	13	20	22																																																																									
5	7	14	16	23																																																																									
Square array 4	Square array 5	Square array 6																																																																											
<table border="1"><tr><td>1</td><td>9</td><td>12</td><td>20</td><td>23</td></tr><tr><td>2</td><td>10</td><td>13</td><td>16</td><td>24</td></tr><tr><td>3</td><td>6</td><td>14</td><td>17</td><td>25</td></tr><tr><td>4</td><td>7</td><td>15</td><td>18</td><td>21</td></tr><tr><td>5</td><td>8</td><td>11</td><td>19</td><td>22</td></tr></table>	1	9	12	20	23	2	10	13	16	24	3	6	14	17	25	4	7	15	18	21	5	8	11	19	22	<table border="1"><tr><td>1</td><td>10</td><td>14</td><td>18</td><td>22</td></tr><tr><td>2</td><td>6</td><td>15</td><td>19</td><td>23</td></tr><tr><td>3</td><td>7</td><td>11</td><td>20</td><td>24</td></tr><tr><td>4</td><td>8</td><td>12</td><td>16</td><td>25</td></tr><tr><td>5</td><td>9</td><td>13</td><td>17</td><td>21</td></tr></table>	1	10	14	18	22	2	6	15	19	23	3	7	11	20	24	4	8	12	16	25	5	9	13	17	21	<table border="1"><tr><td>1</td><td>6</td><td>11</td><td>16</td><td>21</td></tr><tr><td>2</td><td>7</td><td>12</td><td>17</td><td>22</td></tr><tr><td>3</td><td>8</td><td>13</td><td>18</td><td>23</td></tr><tr><td>4</td><td>9</td><td>14</td><td>19</td><td>24</td></tr><tr><td>5</td><td>10</td><td>15</td><td>20</td><td>25</td></tr></table>	1	6	11	16	21	2	7	12	17	22	3	8	13	18	23	4	9	14	19	24	5	10	15	20	25
1	9	12	20	23																																																																									
2	10	13	16	24																																																																									
3	6	14	17	25																																																																									
4	7	15	18	21																																																																									
5	8	11	19	22																																																																									
1	10	14	18	22																																																																									
2	6	15	19	23																																																																									
3	7	11	20	24																																																																									
4	8	12	16	25																																																																									
5	9	13	17	21																																																																									
1	6	11	16	21																																																																									
2	7	12	17	22																																																																									
3	8	13	18	23																																																																									
4	9	14	19	24																																																																									
5	10	15	20	25																																																																									

As a check, the main right diagonal of square array 6 produces the first row of square array 2, i.e., 1,7,13,19,25.

Steps 7 and 8: Let the pair of square arrays be 1 and 2. Then, the first rows are 1,2,3,4,5 and 1,7,13,19,25 to form:

Square 1	Square 2
1 2 3 4 5	1 7 13 19 25
7 8 9 10 6	2 8 14 20 21
13 14 15 11 12	3 9 15 16 22
19 20 16 17 18	4 10 11 17 23
25 21 22 23 24	5 6 12 18 24

Steps 9 and 10: Let the pair of square arrays for these steps be 3 and 4 to form:

Square 3	Square 4
1 8 15 17 24	1 9 12 20 23
9 11 18 25 2	8 11 19 22 5
12 19 21 3 10	15 18 21 4 7
20 22 4 6 13	17 25 3 6 14
23 5 7 14 16	24 2 10 13 16

Steps 11 and 12: The last pair of the $(k+1)/2 = 3$ pairs is square arrays 5 and 6. Repeating the above process results in the following two squares:

Square 5	Square 6
1 10 14 18 22	1 6 11 16 21
6 15 19 23 2	10 15 20 25 5
11 20 24 3 7	14 19 24 4 9
16 25 4 8 12	18 23 3 8 13
21 5 9 13 17	22 2 7 12 17

Following through the above steps produces a balanced lattice square design. Each pair of treatments occurs once with every other treatment in a row and once in a column. Use of steps $k+2, k+4, \dots, 2k+1$ produces a semi-balanced lattice square design where every pair of treatments occurs together once in either a row or a column. The same property holds for the

$(k+1)/2$ squares from steps $k+3, k+5, \dots, 2k+2$. Use of subsets of the $k+1$ squares produces designs with concurrences of pairs of treatments being either zero or one in the rows and also in the columns. In all cases, the designs are optimal in the sense of pairwise balance and also variance balance.

k NOT A PRIME NUMBER

The first $k+1$ steps of the Algorithm always produces $k+1$ squares. The concurrences of pairs of treatments in blocks are not in unity even for prime powers. To illustrate, let $k=4$ to obtain the following five complete blocks:

1	2	3	4	1	5	9	13	1	6	11	16	1	7	9	15	1	8	11	14
5	6	7	8	2	6	10	14	2	7	12	13	2	8	10	16	2	5	12	15
9	10	11	12	3	7	11	15	3	8	9	14	3	5	11	13	3	6	9	16
13	14	15	16	4	8	12	16	4	5	10	15	4	6	12	14	4	7	10	13

In the above design, each treatment appears once with 11 of the other treatments in a block; it appears twice with 2 treatments and zero times with 2 treatments. In a balanced incomplete block design for $v=16$, $b=20$, $k=4$, each treatment would occur once, $\lambda=1$, with every other treatment. Instead of using the first $k+1$ steps of the Algorithm, substitute the $k+1$ arrays for a balanced incomplete block design when k is a prime power. Then, proceed through steps $k+2, k+4, \dots, 2k+1$ and then continue the procedure for $k+1$ steps. The steps transposing a square are omitted. For the rows of square 1 use array 1 and use array 2 for the columns. For the second square use array 2 and array 3 for the columns. The $k+1$ st square uses array $k+1$ for the rows and array 1 for the columns.

This is illustrated for $k=4$. Balanced incomplete blocks for $v=16$, $k=4$ are:

Array 1	Array 2	Array 3	Array 4	Array 5
1 2 3 4	1 5 9 13	1 6 11 16	1 14 7 12	1 8 10 15
5 6 7 8	2 6 10 14	5 2 15 12	13 2 11 8	2 7 9 16
9 10 11 12	3 7 11 15	9 14 3 8	5 10 3 16	3 6 12 13
13 14 15 16	4 8 12 16	13 10 7 4	9 6 15 4	4 5 11 14

Step $k+2=6$: use array 1 as rows and array 2 as columns to obtain square 1:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Steps $k+4$, $k+6$, $k+8$, $k+10$: Use array 2 as rows and array 3 as columns for square 2; use 3 as rows and 4 as columns for square 3; use 4 as rows and 5 as columns to obtain square 4; and use array 5 as rows and array 1 as columns to obtain square $5=k+1$ as follows:

Square 2	Square 3	Square 4	Square 5
1 5 9 13	1 6 11 16	1 7 12 14	1 8 10 15
6 2 14 10	7 4 13 10	8 2 13 11	2 7 9 16
11 15 3 7	12 15 2 5	10 16 3 5	3 6 12 13
16 12 8 4	14 9 8 3	15 9 6 4	4 5 11 14

The above procedure produces a balanced lattice square design for k a prime power, with every pair of treatments occurring once in a row and once in a column.

When k is not a prime power, a balanced lattice square design is not achievable. If there exists t pairwise orthogonal Latin squares of order k , then one may use these to form $t+1$ incomplete block arrays which will have treatments pairs occurring zero or one time in the

$k(t+1)=b$ blocks. Then, the $t+1$ arrays can be used as described above for the prime power case for $t+1$ squares.

As an alternative, the Algorithm of the preceding section may be used directly to obtain $k+1$ arrays. For example, the $k+1=7$ arrays obtained for $v=36$ are:

Array 1

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Array 2

1	7	13	19	25	31
2	8	14	20	26	32
3	9	15	21	27	33
4	10	16	22	28	34
5	11	17	23	29	35
6	12	18	24	30	36

Array 3

1	8	15	22	29	36
2	9	16	23	30	31
3	10	17	24	25	32
4	11	18	19	26	33
5	12	13	20	27	34
6	7	14	21	28	35

Array 4

1	9	17	19	27	35
2	10	18	20	28	36
3	11	13	21	29	31
4	12	14	22	30	32
5	7	15	23	25	33
6	8	16	24	26	34

Array 5

1	10	13	22	25	34
2	11	14	23	26	35
3	12	15	24	27	36
4	7	16	19	28	31
5	8	17	20	29	32
6	9	18	21	30	33

Array 6

1	11	15	19	29	33
2	12	16	20	30	34
3	7	17	21	25	35
4	8	18	22	26	36
5	9	13	23	27	31
6	10	14	24	28	32

Array 7

1	12	17	22	27	32
2	7	18	23	28	33
3	8	13	24	29	34
4	9	14	19	30	35
5	10	15	20	25	36
6	11	16	21	26	31

Note that a more pairwise balanced arrangement could be obtained by using the first rows of arrays 4, 5, 6, and 7, as (1,9,17,20,27,35), (1,10,14,23,26,34), (1,11,16,21,30,33), and (1,12,18,24,28,32), respectively. These were obtained such that treatment 1 occurred once

with every other treatment. This results in concurrences of pairs of treatments as 0, 1, or 2.

The seven squares for a lattice square design are obtained from arrays i and $i+1$, $i=1, \dots, 7$. These are:

Square 1(arrays 1 and 2)

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Square 2(arrays 2 and 3)

1	7	13	19	25	31
8	14	20	26	32	2
15	21	27	33	3	9
22	28	34	4	10	16
29	35	5	11	17	23
36	6	12	18	24	30

Square 3(arrays 3 and 4)

1	8	15	22	29	36
9	16	23	30	31	2
17	24	25	32	3	10
19	26	33	4	11	18
27	34	5	12	13	20
35	6	7	14	21	28

Square 4(arrays 4 and 5)

1	9	17	19	27	35
10	18	20	28	36	2
13	21	29	31	3	11
22	30	32	4	12	14
25	33	5	7	15	23
34	6	8	16	24	26

Square 5(arrays 5 and 6)

1	10	13	22	25	34
11	14	23	26	35	2
15	24	27	36	3	12
19	28	31	4	7	16
29	32	5	8	17	20
33	6	9	18	21	30

Square 6(arrays 6 and 7)

1	11	15	19	29	33
12	16	20	30	34	2
17	21	25	35	3	7
22	26	36	4	8	18
27	31	5	9	13	23
32	6	10	14	24	28

Square 7(arrays 7 and 1)

1	12	17	22	27	32
2	7	18	23	28	33
3	8	13	24	29	34
4	9	14	19	30	35
5	10	15	20	25	36
6	11	16	21	26	31

If only two replicates were desired, use squares 1 and 3; if three use 1, 3, and 5; and if four use 1, 3, 5, and 7. As stated above, a better design in the sense of better pairwise and variance balance would be obtained by using the substituted arrays given above. The procedure can be used for any value of k .

VARIETY CUTTING TO OBTAIN DESIGNS

FOR $v < k^2$

For field or other experiments where the rows and columns in a complete block of a lattice square design are laid out spatially, or through time, any number of varieties may be deleted. The spaces where the varieties would have been can remain blank, be planted with a filler or other treatments can be included in one of the blank spaces to obtain an augmented lattice square design. To illustrate, let $k=2$. Then, the $3+1=4$ squares are:

1	2	3
4	5	6
7	8	9

1	4	7
2	5	8
3	6	9

1	5	9
6	7	2
8	3	4

1	6	8
5	7	3
9	2	4

Let $v=7$ and let there be 8 other treatments (a,b,c,d,e,f,g) each to be included once, resulting in the four squares:

1	2	3
4	5	6
7	a	b

1	4	7
2	5	c
3	6	d

1	5	f
6	7	2
e	3	4

1	6	g
5	7	3
h	2	4

The rows and columns of each square would be randomized in the experiment.

ANALYSIS

The usual linear model for a lattice square design is:

$$Y_{ghij} = \mu + \beta_g + \rho_{gh} + \gamma_{gi} + \tau_j + \varepsilon_{ghij},$$

where μ is an effect common to all experimental units, β_g is the g th complete block effect, ρ_{gh} is the h th row within the g th block effect, γ_{gi} is the i th column within the g th block effect, τ_j is the j th treatment effect, ε_{ghij} is a random error component with mean zero and variance σ_ε^2 , $g=1,2,\dots,r$; $h=1,2,\dots,k$; $i=1,2,\dots,k$; $j=1,2,\dots,v$, ρ_{gh} is a random effect with mean zero and variance σ_ρ^2 , and γ_{gi} is a random effect with mean zero and variance σ_γ^2 . Analysis with recovery of row and column information is suggested.

REFERENCES

Federer, W. T. 1955

Experimental Design-Theory and Application, Macmillan, New York

(Reprinted by the Oxford and IBH Publishing Co. 1967, 1974).

Khare, M. and W.T. Federer (1981)

A simple construction procedure for resolvable incomplete block designs for any number of treatments, *Biometrics Journal* 23 (2),

121-132.

Federer, W.T. and B.L. Raktoe (1965)

General theory of prime power lattice designs. Lattice rectangles for $v=s^m$ treatments in s^r rows and s^c columns for $r+c=m$, $r \neq c$ and $v < 1000$,

J. American Statistical Association 60, 891-904.