

Your quick start guide to



CycDesignN

Block and row-column designs

Table of Contents

What is CycDesignN?	2
Some terminology	2
Basic block designs	2
Resolvable block design	2
Non-resolvable block design	3
Basic row-column designs	4
Resolvable row-column design	4
Non-resolvable row-column design	4
Efficiency	5
Worked examples: Steps to generate the designs shown in the examples	5
Worked Example 1: Resolvable block design.....	5
Worked Example 2: Non-resolvable block design.....	10
Worked Example 3: Resolvable row-column design	12
Worked Example 4: Non-resolvable row-column design.....	16
Latinizing.....	19
Basic Latinizing	19
Doubly resolvable block design.....	24
Latinizing row-column designs.....	28
Explanation of design generation options and buttons in CycDesignN	31
Where are my files stored?	31
Changing file names	31
Changing the location of saved files.....	31
Working directory	32
Log file	32
Design file.....	32
Help	33
Latinizing	33
Factorial or nested treatments	33
Single factor with the levels grouped and nested	33
Factorial.....	35
Unequal replication.....	36
Spatial.....	36
Seeds	36
Two stage/One stage	37
Blocks in columns.....	37
CycAnalysis.....	38

What is CycDesignN?

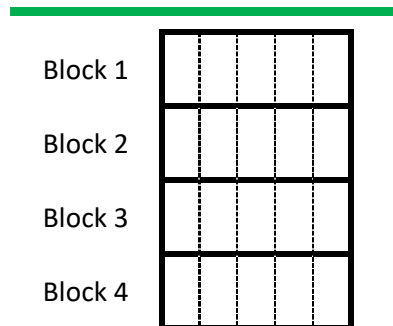
CycDesignN is a comprehensive program for generating trial designs. There are 4 main classes of design:

- Resolvable
- Non-resolvable
- Partially replicated
- Cross-over

The first 3 can be either Block designs or Row-Column designs. This guide focuses on Resolvable and Non-resolvable Block and Row-Column designs.

Some terminology

In this guide, *Plot* can mean anything that you apply a treatment to, including plants in a pot, individuals or groups of people or animals, and so on. A *Unit* in CycDesignN is another name for *Plot*. A *Block* is a group of plots, usually chosen because the plots in the block are in some way similar, with different blocks in some way differing. For example, if the trial is a greenhouse pot trial, pots closest to the glasshouse wall are likely to have more light than those further away, so the plots could be grouped into blocks by distance from the glasshouse wall, indicated by the green line in the following image.



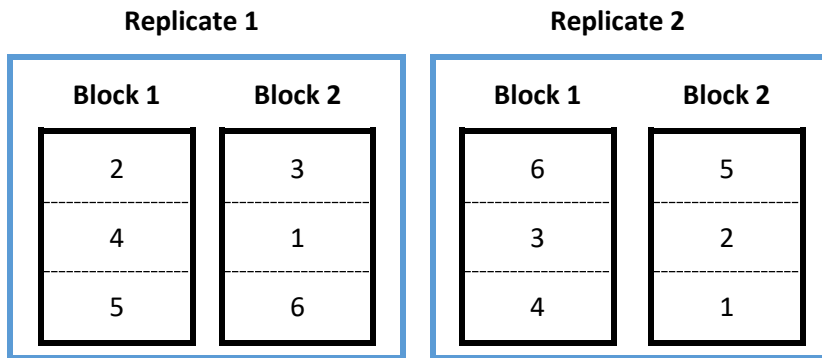
If the plots in a block are in a row (or column), then the set of plots can be called a row (or column).

Basic block designs

A basic block design has more than one block of plots, which are equally sized, containing k plots per block, and with k less than the number of treatments v . Each of the v treatments is replicated r times. Each block contains a subset of the treatments. There are two types, Resolvable and Non-resolvable.

Resolvable block design

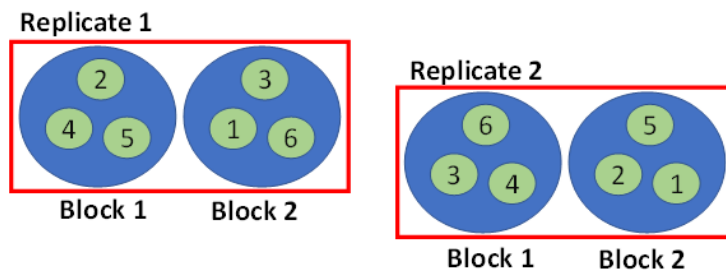
In a resolvable block design the blocks can be grouped together such that each group contains a complete set of treatments. There are r groups of blocks, with each group containing just one plot of each treatment, with all treatments appearing in the group. Each group is thus a complete *Replicate*.



Example 1: A resolvable block design with 6 treatments ($v=6$, numbered 1-6 in the diagram), each replicated twice ($r=2$), with blocks of size 3 ($k=3$).

Worked example 1 (page 5) explains the steps to generate this resolvable block design.

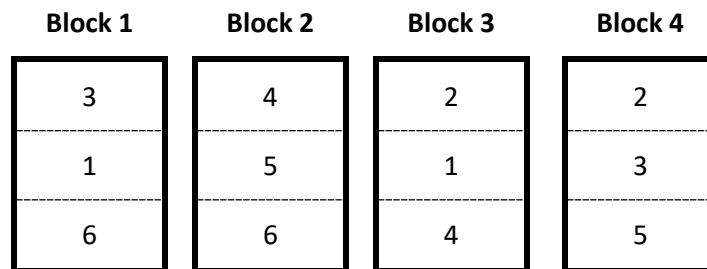
Note that the plots do not need to be arranged in a rectangular grid. A block may be a pot, with plots as plants within a pot (see next image), or a block may be a day, with the plots as three successive times within each day, and so on.



An alternative arrangement of plots using the same design as in Example 1. Red rectangles are glasshouse benches, blue circles are pots and green circles represent plants.

Non-resolvable block design

In a non-resolvable block design the blocks cannot be grouped together such that each group contains a complete set of treatments.



Example 2: A non-resolvable block design, with 6 treatments ($v=6$), each replicated twice ($r=2$), with 4 blocks of size 3 ($k=3$). It is not possible to pair the blocks such that the pair of blocks contains each treatment exactly once.

For some designs a treatment may appear more than once in a block.

Worked example 2 (page 10) explains the steps to generate this non-resolvable block design.

Somewhat counter-intuitively, the most widely used block design, a randomized block design, can be created using a non-resolvable (rather than resolvable) design. For 6 treatments, replicated twice, a randomized block design would be generated for blocks of size 6, with 2 blocks.

Basic row-column designs

As the name suggests, the plots are arranged in a two-dimensional array (or arrays) of plots, with number of rows nr and number of columns nc per array. As with block designs, the array need not be a rectangular layout. For example, plots may be a run of a machine, rows may be days and columns 3 people carrying out a run each in each day.

Resolvable row-column design

Each of the r replicates is laid out as the number of rows nr and number of columns nc . There are thus $r nr \times nc$ arrays of plots, with each array containing a full set of the treatments, with only one plot of each treatment per array. Therefore, $nr \times nc$ equals the number of treatments v . Example 3 shows the layout for a trial with 2 replicates of 12 treatments, with the plots for each replicate arranged in 4 rows by 3 columns.

Replicate 1			Replicate 2		
8	5	3	5	7	12
12	1	10	11	8	2
6	2	7	10	9	4
9	4	11	6	1	3

Example 3: A resolvable row-column design, with 12 treatments ($v=12$), each replicated twice ($r=2$), arranged in two 4×3 arrays ($nr=4$; $nc=3$)

Worked example 3 (page 12) explains the steps to generate this resolvable row-column design.

Non-resolvable row-column design

This is a single array containing all $v \times r$ plots. The array has nr rows and nc columns. Thus, $nr \times nc$ is equal to the number of treatments v times the number of replicates r . With this design, depending on the numbers of rows and columns, it is possible to get a treatment appearing more than once within a given row or column.

9	4	10	8	12	7
7	2	12	1	6	5
11	9	2	5	8	3
1	6	3	10	11	4

Example 4: A non-resolvable block design, with twelve treatments ($v=12$), each replicated twice ($r=2$), with the 24 plots arranged as four rows ($nr=4$) and six columns ($nc=6$).

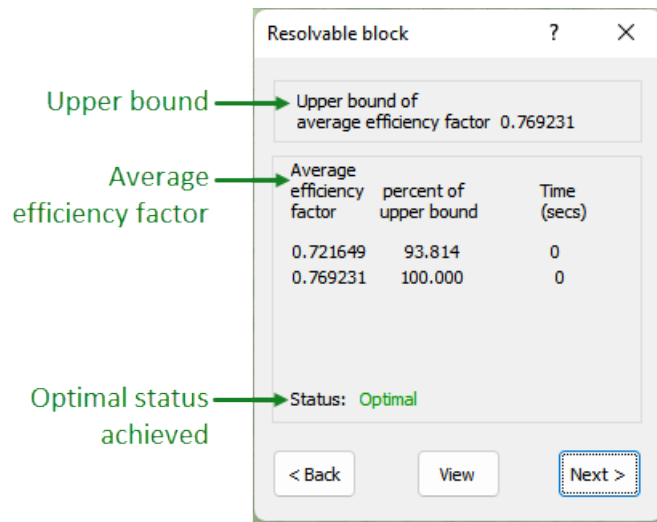
Worked example 4 (page 16) explains the steps to generate this non-resolvable row-column design.

For some designs, a treatment may appear more than once in a row or column, although in general it is better to employ blocking structures that avoid non-binary treatments.

The most widely known and simplest row-column design is a Latin square. This can be generated using a non-resolvable design, with nr and nc equalling the number of treatments v . Designs that are derived from a Latin square, such as an incomplete Latin square (Latin square with one row or column deleted) or extended Latin square (has an extra row or column) are also non-resolvable row-column designs.

Efficiency

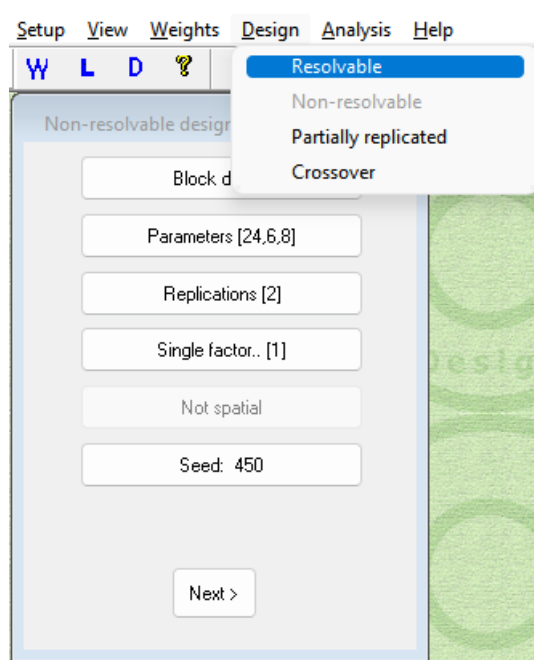
CycDesigN generates designs to maximise *Efficiency*. Taking into account the trial design, efficiency is a measure of the precision of a comparison between two treatments (a treatment pair). Average efficiency is the mean over all treatment pairs of the estimated precision. Efficiency has a maximum value of 1. A randomized block design has an efficiency of 1 for every possible pair of treatments so has the maximum average efficiency 1. CycDesigN estimates a theoretical upper bound (limit) for the average efficiency. An iterative process is used to maximise the average efficiency. At each iteration the average efficiency is shown, as this is the average as a percentage of the upper bound. A design where the average efficiency equals the upper bound is 'Optimal'. However, it is fairly common for the iterative process to never reach optimality. This can be because the estimated upper bound is an over-estimate of the true maximum value for a design.



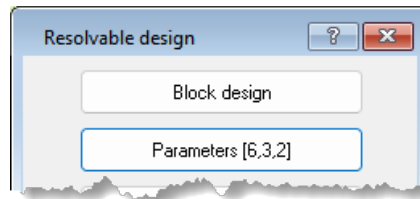
Worked examples: Steps to generate the designs shown in the examples

Worked Example 1: Resolvable block design

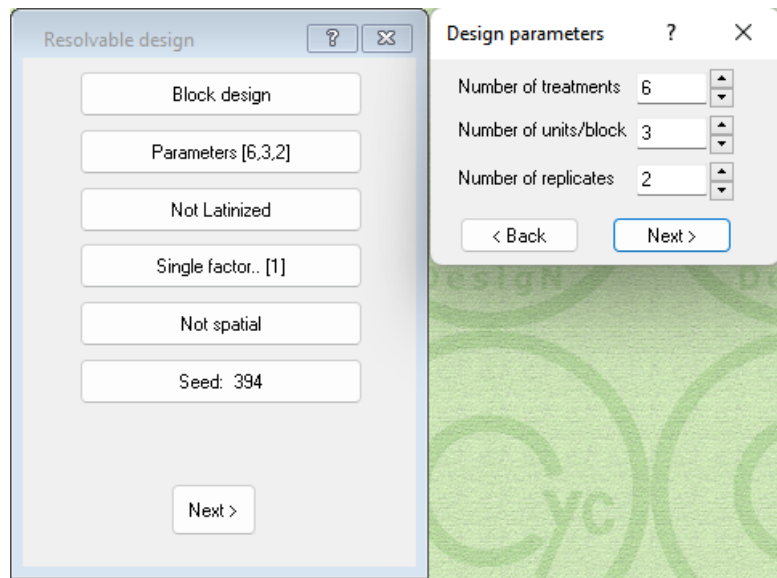
1. From the main menu click **Design** then select **Resolvable**.



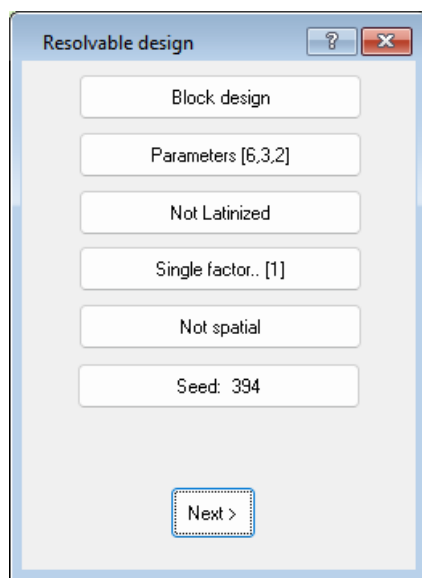
2. Toggle the top button so that **Block design** is displayed.



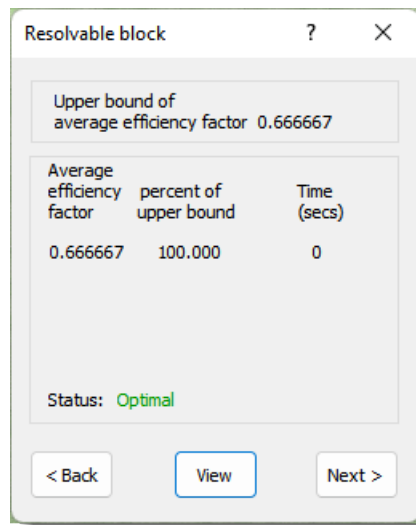
3. Click **Parameters** and enter the *Number of treatments* (6), *Number of units/block* (3) and *Number of replicates* (2).



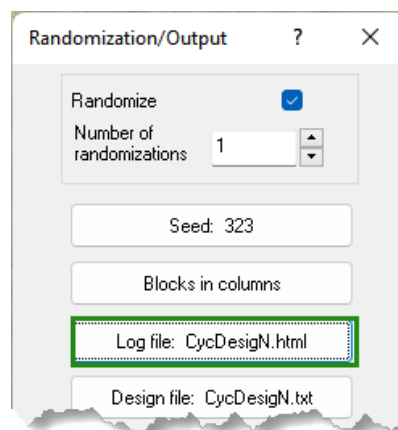
4. Click **Next** to close the *Design parameters* menu then click **Next** on the *Resolvable design* menu to begin the iterative search for an optimal design.



- An optimal arrangement for this very simple design is reached immediately. In general, there will be several iterations with the Average efficiency factor changing at each iteration. Click **Next** to move to randomization of the design.



- Note the names on the **Log file** and **Design file** buttons. These default names are used every time you generate a new design so we highly recommend you enter a new name to avoid overwriting a previously saved design. Click **Log file** and type 'resolvable_block' then click **Save**.



- Click **Next** to perform the randomization and generate the design.

The CycDesignN Log file will open automatically. The top section of the file displays a summary of the generated design.

```

CycDesignN logfile

Date and time: Fri Sep  2 15:39:35 2022

Design properties
  Resolvable
  Block design
  Single factor

Design parameters
  Number of treatments           =    6
  Number of units/block         =    3
  Number of blocks               =    2
  Number of replicates          =    2

Random number seed for design generation = 394

Average efficiency factors (Upper bounds)
  Block      0.666667  (0.666667)

Concurrencies

  Concurrence      Block
    0                5
    1                8
    2                2

Random number seed for design randomization = 323

```

The bottom section of the file shows further information about the generated design (*Concurrencies*), and a diagram showing the generated layout.

```

Concurrencies

  Concurrence      Block
    0                5
    1                8
    2                2

Random number seed for design randomization = 323

Randomized design 1

block  +-----+
plot   +-----+
  1 |  2  3
  2 |  4  1
  3 |  5  6

  1 |  6  5
  2 |  3  2
  3 |  4  1

```

For our design with 6 treatments, there are 15 possible pairs of treatments (Treatments 1,2; 1,3; 1,4 ... 5,6).

In the Log file, *Concurrencies* summarises how many times each of the 15 pairs occurs within a block. For example, the pair 1,4 does not occur at all and the pair 3,6 occurs twice. Concurrencies show that five pairs do not occur at all (Concurrence=0), 8 appear once, and two twice. Ideally, each pair would occur equally often, but this is frequently not possible.

```

Concurrencies

Concurrence      Block
      0             5
      1             8
      2             2

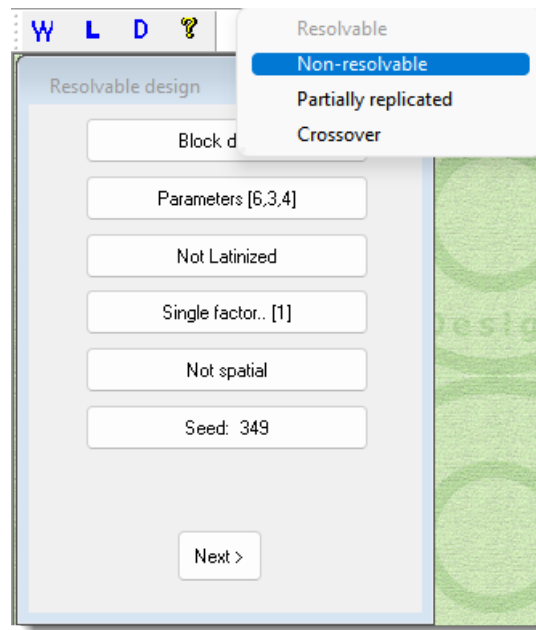
Random number seed for design randomization = 323
  
```

The file *CycDesigN.csv* is stored in your working directory - the default location is *C:\ProgramData\DATA*. Column A shows the replicate number, B the block number within each replicate, C the plot number within each Block, and D the treatment to apply to each plot. This information can be used to generate a skeleton data collection sheet.

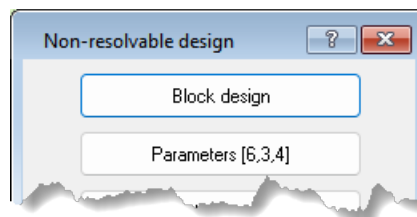
	A	B	C	D	E
1	1	1	1	2	
2	1	1	2	4	
3	1	1	3	5	
4	1	2	1	3	
5	1	2	2	1	
6	1	2	3	6	
7	2	1	1	6	
8	2	1	2	3	
9	2	1	3	4	
10	2	2	1	5	
11	2	2	2	2	
12	2	2	3	1	
13					

Worked Example 2: Non-resolvable block design

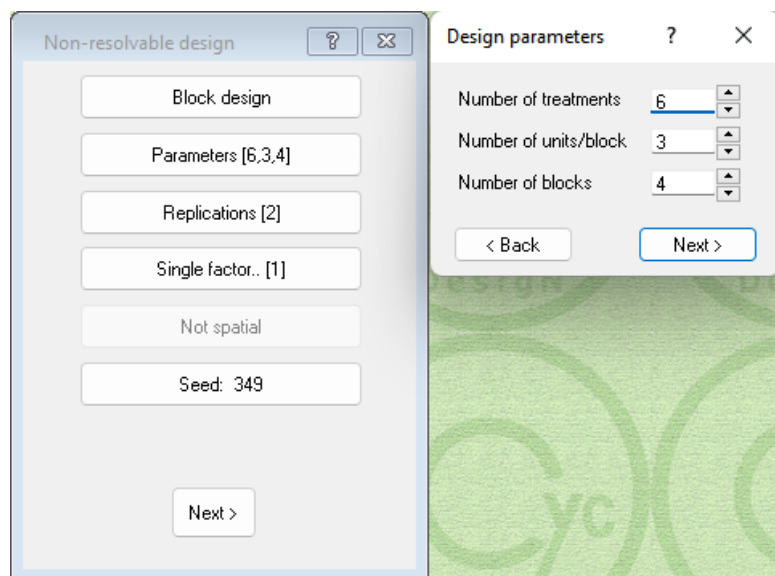
1. From the main menu click **Design** then select **Non-resolvable**.



2. Toggle the top button so that **Block design** is displayed.



3. Click **Parameters** and enter the *Number of treatments* (6), *Number of units/block* (3) and *Number of replicates* (4).



4. Click **Next** to close the *Design parameters* menu then click **Next** on the underlying menu to begin the iterative search for an optimal design.
5. Wait for the process to converge (become Optimal) then click **Next**.

6. Click **Log file** and type the name 'non-resolvable_block' then click **Save**.

7. Click **Next** to generate the design.

The CycDesigN Log file will open automatically. The top section of the file displays a summary of the generated design. Notice that the Average efficiency for this non-resolvable design is greater than that for the resolvable design of Example 1, with the same number of treatments, block size, and number of replications.

```
CycDesigN logfile
Date and time: Mon Sep  5 08:26:33 2022

Design properties
Non-resolvable
Block design
Single factor

Design parameters
Number of treatments           = 6
Number of units/block         = 3
Number of blocks               = 4
Number of replications         = 2

Random number seed for design generation = 349

Average efficiency factors (Upper bounds)
Block      0.769231 (0.769231)

Concurrencies

Concurrence      Block
0                 3
1                 12
```

In the bottom section of the file notice that the list of concurrencies is more equal than for the resolvable design of Example 1 where 2 treatment pairs occurred twice and 5 not at all.

```
Concurrencies

Concurrence      Block
0                 3
1                 12

Random number seed for design randomization = 249

Randomized design 1

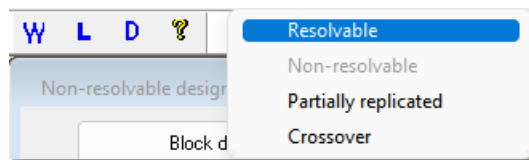
+-----+
block      1   2   3   4
plot      +-----+
1 |   3   4   2   2
2 |   1   5   1   3
3 |   6   6   4   5
```

The file `CycDesigN.csv` contains the generated factors and is stored in your working directory - the default location is `C:\ProgramData\DATA`. Column A shows the block number, B the plot number within each Block, and C the treatment to apply to each plot.

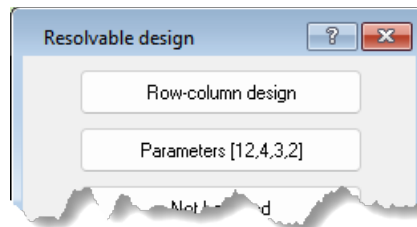
	A	B	C	D
1	1	1	3	
2	1	2	1	
3	1	3	6	
4	2	1	4	
5	2	2	5	
6	2	3	6	
7	3	1	2	
8	3	2	1	
9	3	3	4	
10	4	1	2	
11	4	2	3	
12	4	3	5	
13				

Worked Example 3: Resolvable row-column design

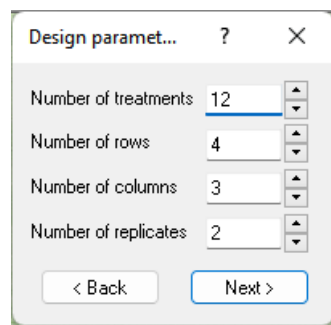
- From the main menu click **Design** then select **Resolvable**.



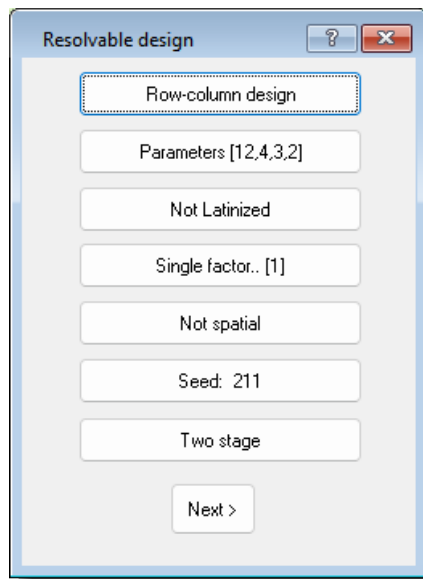
- Toggle the top button so that **Row-column** design is displayed.



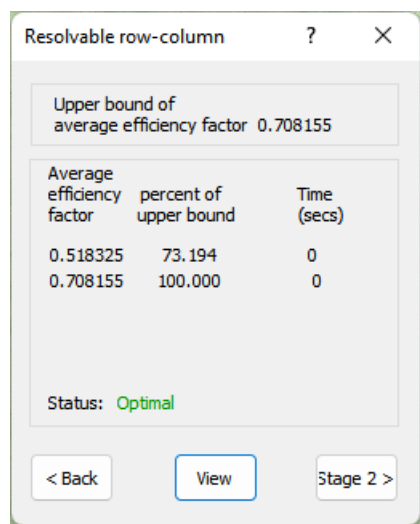
- Click **Parameters** and enter the *Number of treatments* (12), *Number of rows* (4), *Number of columns* (3) and *Number of replicates* (2). Then click **Next**.



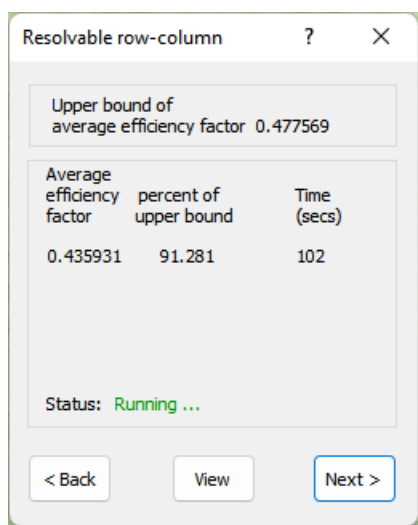
4. By default, row-column designs are generated in two stages. For this example, the two-stage method is used. (You can change this to a single stage by toggling the **Two stage** button so that **One stage** displays.) Click **Next** to begin the iterative search for an optimal design.



5. The iterative process for the first stage takes just two steps to reach the optimal Average efficiency. Click **Stage 2 >**.

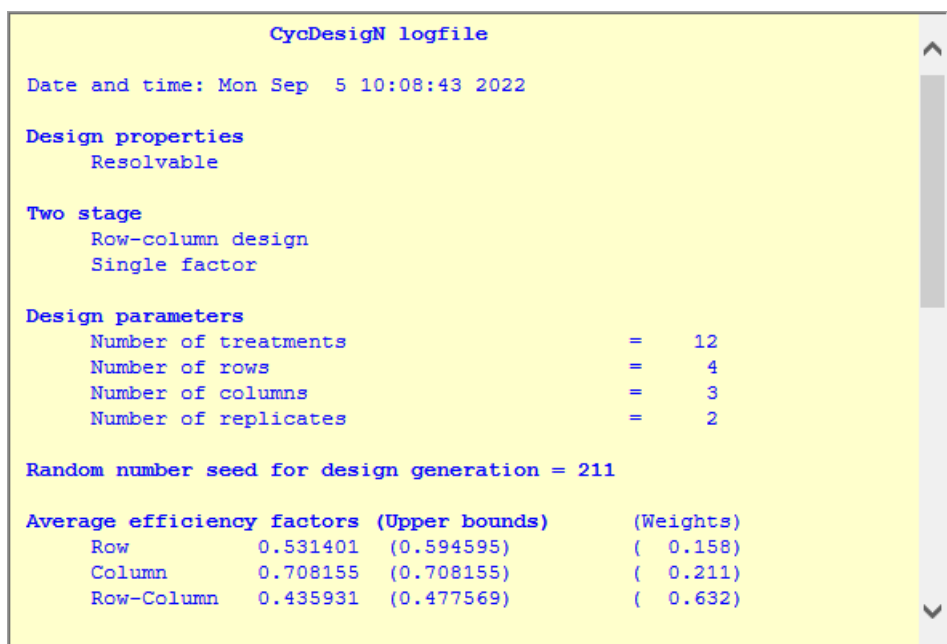


- The second stage of iterations does not reach the optimal value very quickly. At this stage, you can decide to continue with a design that has sub-optimal average efficiency or wait to see if there is any improvement. Click **Next** to move to randomization of the design.



- As with the previous designs the final step is randomization. You can click **Log file** and enter a new name or just click **Next**. The CycDesign Log file will open automatically.

The following image shows the top section of the Log file with a summary of the generated design.



The bottom section of the file contains the concurrencies and a diagram of the generated layout.

```

Concurrencies

Concurrence      Row      Column
      0           43         33
      1           22         30
      2            1          3

Random number seed for design randomization = 323

Randomized design 1

column +-----+
      1   2   3
row +-----+
  1 |   8   5   3
  2 |  12   1  10
  3 |   6   2   7
  4 |   9   4  11

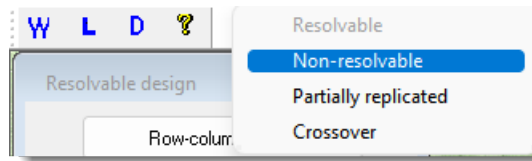
  1 |   5   7  12
  2 |  11   8   2
  3 |  10   9   4
  4 |   6   1   3
  
```

The file `CycDesigN.csv` contains the generated factors. Column A shows the replicate number, B the column number within each replicate, C the row number within each replicate, and D the treatment to apply to each plot. This information can be used to generate a skeleton data collection sheet.

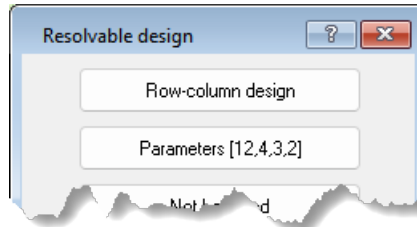
	A	B	C	D	E
1	1	1	1	8	
2	1	1	2	12	
3	1	1	3	6	
4	1	1	4	9	
5	1	2	1	5	
6	1	2	2	1	
7	1	2	3	2	
8	1	2	4	4	
9	1	3	1	3	
10	1	3	2	10	
11	1	3	3	7	
12	1	3	4	11	
13	2	1	1	5	
14	2	1	2	11	
15	2	1	3	10	
16	2	1	4	6	
17	2	2	1	7	
18	2	2	2	8	
19	2	2	3	9	
20	2	2	4	1	
21	2	3	1	12	
22	2	3	2	2	
23	2	3	3	4	
24	2	3	4	3	
25					

Worked Example 4: Non-resolvable row-column design

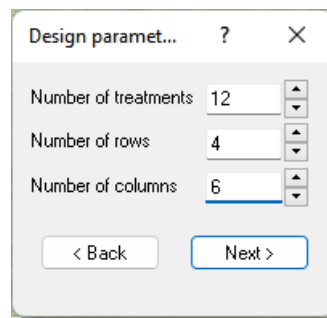
1. From the main menu click **Design** then select **Non-resolvable**.



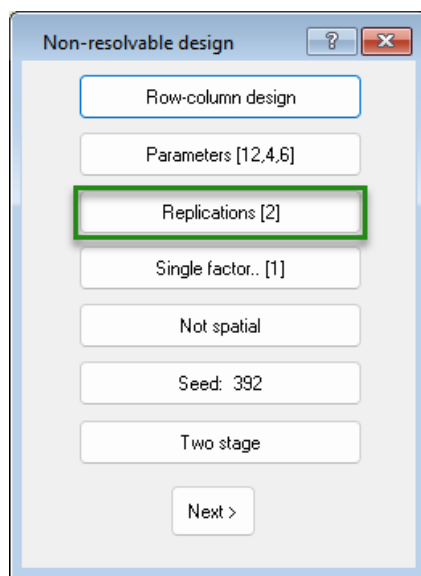
2. Toggle the top button so that **Row-column design** is displayed.



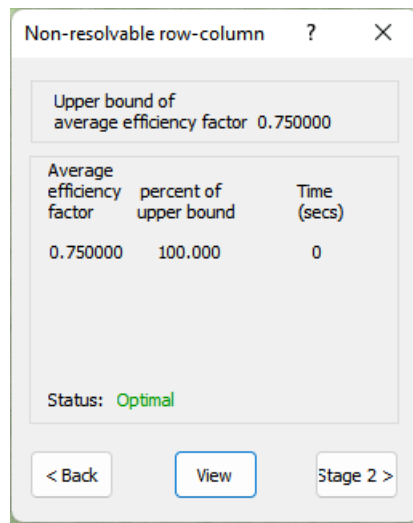
3. Click **Parameters** and enter the *Number of treatments* (12), *Number of rows* (4) and *Number of columns* (6).



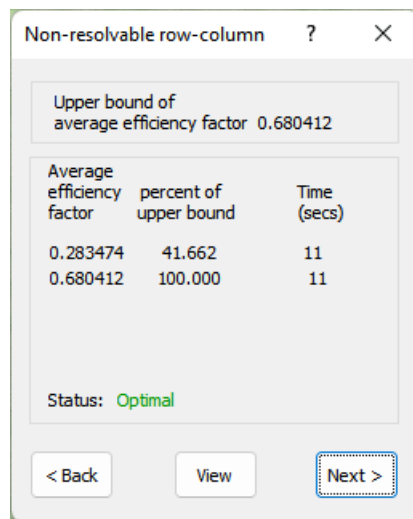
4. Note that CycDesigN works out the number of replicates (2) from the numbers of treatments, rows and columns. By default the design will be generated in two stages. Click **Next** to start the iterative search for an optimal design.



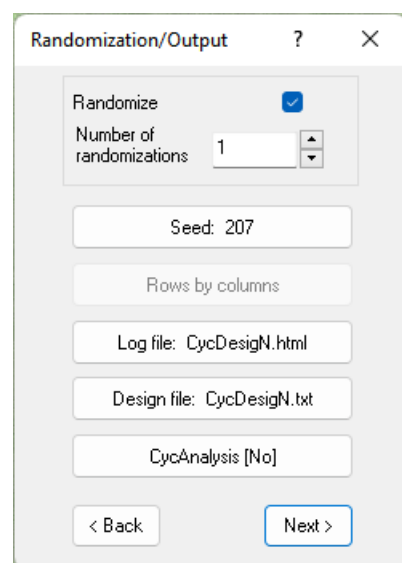
- The iterative process for the first stage takes just one step to reach the optimal average efficiency. Click **Stage 2 >**.



- The second stage of iterations reaches the optimal value very quickly. Click **Next**.



- As with the previous examples, the final step is randomization. You can click **Log file** and enter a new name or just click **Next** to finish the process and open the Log file.



The top section of the Log file summarizes the generated design while the bottom section contains the Concurrencies and a diagram of the generated layout.

```

CycDesignN logfile

Date and time: Mon Sep  5 13:04:57 2022

Design properties
  Non-resolvable

Two stage
  Row-column design
  Single factor

Design parameters
  Number of treatments           = 12
  Number of rows                 =  4
  Number of columns              =  6
  Number of replications         =  2

Random number seed for design generation = 392

Average efficiency factors (Upper bounds)      (Weights)
Row      0.880000 (0.880000) ( 0.176)
Column  0.750000 (0.750000) ( 0.118)
Row-Column 0.680412 (0.680412) ( 0.706)

Concurrencies

Concurrence      Row      Column
0                 12      30
1                 48      36
2                 6       0

Random number seed for design randomization = 207

Randomized design 1

-----
column  +-----+-----+-----+-----+-----+
        | 1     2     3     4     5     6
row     +-----+-----+-----+-----+-----+
1 |      9     4    10     8    12     7
2 |      7     2    12     1     6     5
3 |     11     9     2     5     8     3
4 |      1     6     3    10    11     4
-----

```

The file `CycDesignN.csv` contains the generated factors. A shows the column number, B the row number, and C the treatment to apply to each plot. This information can be used to generate a skeleton data collection sheet.

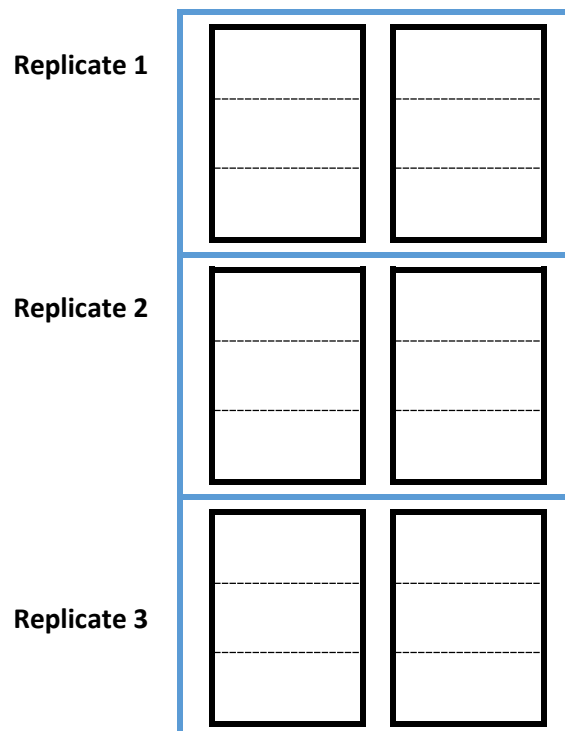
	A	B	C	D
1	1	1	9	
2	1	2	7	
3	1	3	11	
4	1	4	1	
5	2	1	4	
6	2	2	2	
7	2	3	9	
8	2	4	6	
9	3	1	10	
10	3	2	12	
11	3	3	2	
12	3	4	3	
13	4	1	8	
14	4	2	1	
15	4	3	5	
16	4	4	10	
17	5	1	12	
18	5	2	6	
19	5	3	8	
20	5	4	11	
21	6	1	7	
22	6	2	5	
23	6	3	3	
24	6	4	4	

Latinizing

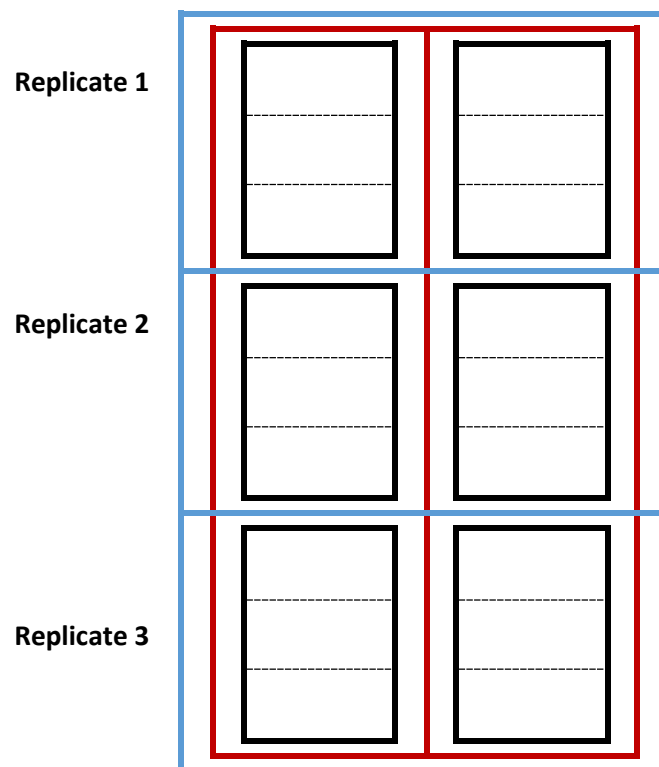
This section describes *Latinizing*, which allows extra blocking to be added to a design. For resolvable designs, latinizing allows further blocking across the replicates, including creating doubly-resolvable designs.

Basic Latinizing

The following example uses 2 replicates ($r=2$) of 6 treatments ($v=6$), with blocks of size 3 ($k=3$). The 18 plots for this trial can be arranged in various ways. Here, they are arranged in a 9x2 grid, with replicates in 3 rows of 2 plots.



Given that it is expected there will be a left to right trend, it would be desirable to make sure that all 6 treatments appear at least once in the left column of plots (below, red outline) and, consequently, also in the right-hand column. This can be achieved with latinizing.



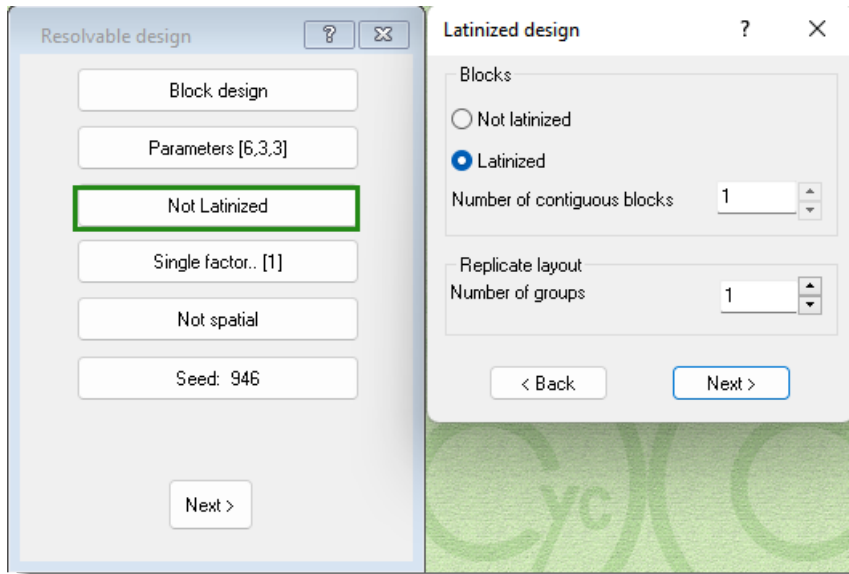
1. From the main menu click **Design** and select **Resolvable**.
2. Toggle the top button so that **Block design** is displayed.
3. Click **Parameters** and enter the values as shown then click **Next**.

The screenshot shows a dialog box titled 'Design parameters' with a question mark and a close button. It contains three input fields with up and down arrows:

- Number of treatments: 6
- Number of units/block: 3
- Number of replicates: 3

At the bottom, there are two buttons: '< Back' and 'Next >'.

4. Click **Not Latinized/Latinized** and select *Latinized* then click **Next** to close this menu.



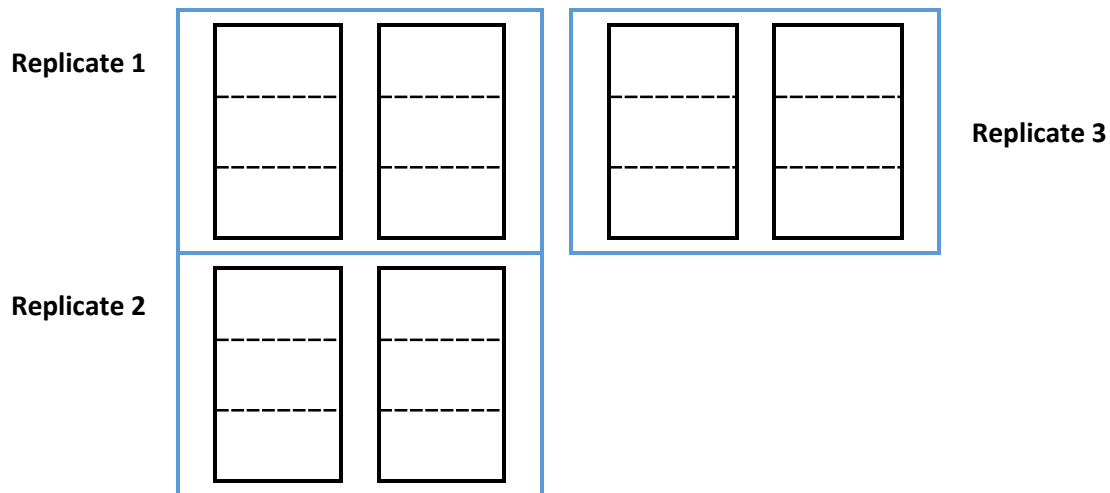
5. Click **Next** on the subsequent menus to begin generating the design.

The resulting design will now have the extra layout feature.

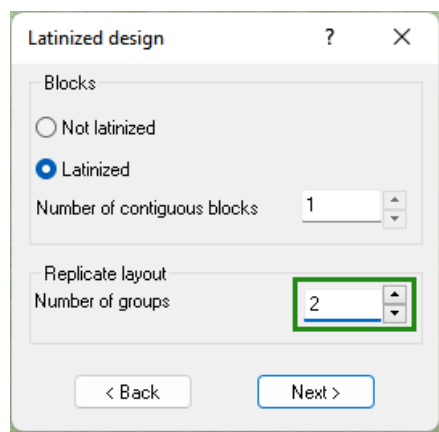
Replicate 1	4	3
	5	2
	1	6
Replicate 2	6	5
	2	1
	4	3
Replicate 3	5	1
	3	2
	6	4

Treatments 4, 5 and 6 appear twice in the left column, with 1, 2 and 3 only once, whilst treatments 1, 2 and 3 appear twice on the right, with 4, 5 and 6 only once.

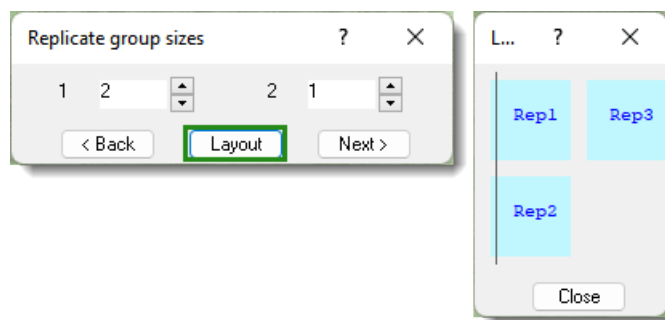
The figure below shows an alternative layout for the replicates. The Latinizing button allows for this, whilst also making sure the treatments are evenly allocated to the left and right columns of replicates 1 and 2. In this diagram, there are two groups of replicates, Replicates 1 and 2, and Replicate 3 in a separate group.



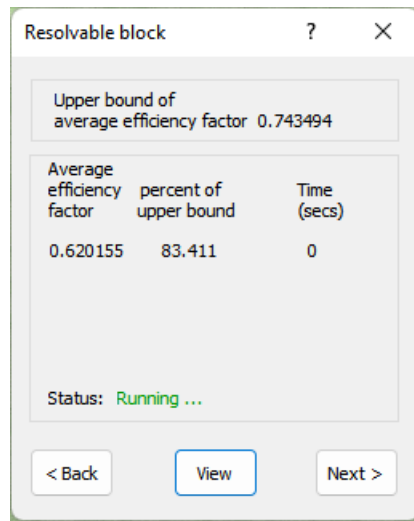
1. To create this arrangement, after clicking **Not Latinized/Latinized** select *Latinized* as before and change *Number of groups* to 2.



2. Click **Next** to open the *Replicate group sizes* menu.
3. Check that the replicates are to be grouped as 2 (Reps 1 and 2) and 1 (Rep 3) then click **Layout**. This opens a little diagram of the arrangement of the replicates, and also shows the Latinization, with a black line. Note that there will be just one black line to indicated that the Latinization process will be carried out in each column.



- Click **Close** on the diagram then click **Next** twice. For this design, the initial Average efficiency factor is only 83% of the theoretical upper bound. Even after 30 minutes a second iteration is not completed. This a common occurrence with CycDesignN, and particularly so for Latinized designs, especially with relatively small numbers of treatments to replicates. Often you can just click **Next**, and the design generated will satisfy the main blocking and Latinization aims. This is the case here.



- Click **Next** then click **Next** again in the *Randomization/Output* menu to produce the Log file.

```

CycDesignN logfile

Date and time: Tue Sep  6 12:21:27 2022

Design properties
  Resolvable
  Block design
  Single factor

Design parameters
  Number of treatments           = 6
  Number of units/block         = 3
  Number of blocks               = 2
  Number of replicates          = 3

Latinized
Number of long column groups = 2
Group  1      2 Replicates
Group  2      1 Replicates

Random number seed for design generation = 946

Average efficiency factors (Upper bounds)
Block      0.620155  (0.743494)

```


The two 'long columns' across replicates 1 and 2 each contain treatments 1-6, forming extra replicates. These can be included as a factor in the analysis.

```

Concurrencies

      Concurrence      Block
        0             5
        1             4
        2             4
        3             2

Random number seed for design randomization = 801

Randomized design 1

-----+-----
block  |          1  2          1  2
plot   |-----+-----
  1 |  1  5  2  5
  2 |  3  4  4  1
  3 |  2  6  6  3

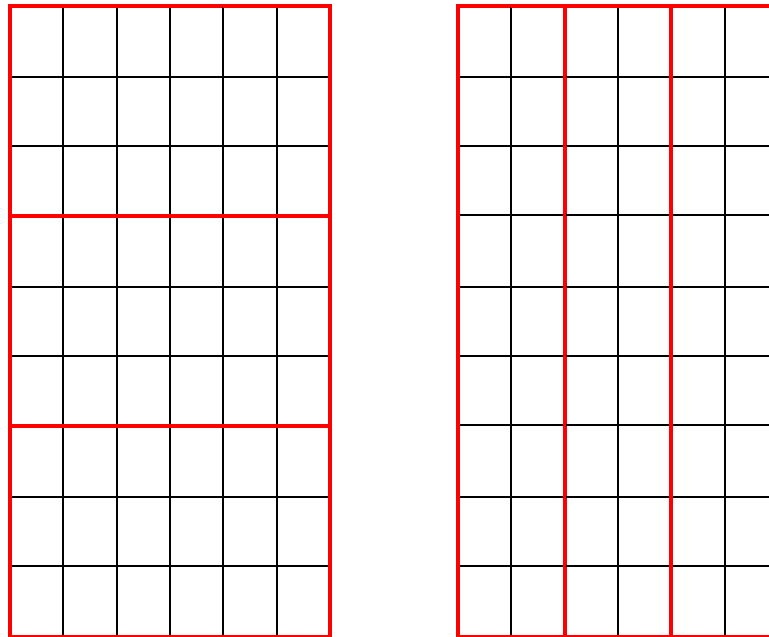
  1 |  6  1
  2 |  4  3
  3 |  5  2

```

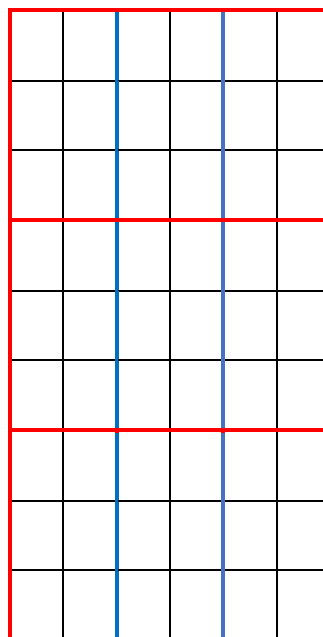
Doubly resolvable block design

For this example, there were 3 replicates of 18 treatments, giving 54 plots. These could be laid out in 9 rows of 6 columns.

With this arrangement, full replicates can be laid out as 3 rows of 6 columns OR 9 rows x 2 columns:

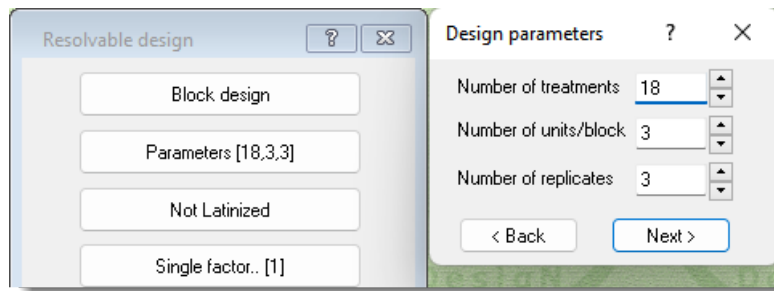


It is possible to arrange the treatments so that both of these replicate arrangements are present within a single layout, giving a *doubly-resolvable* design.

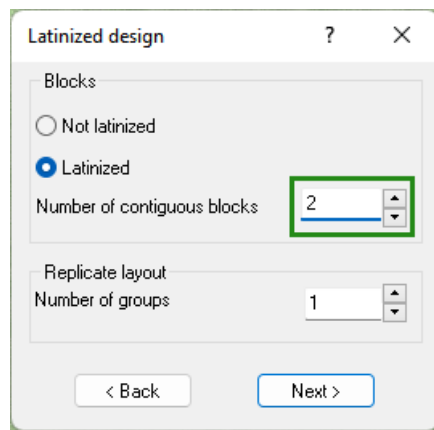


This can be achieved by generating a design for $r=3$ replicates of $v=18$ treatments in blocks of size $k=3$, with the replicates arranged as above using *t-Latinization*, where t is 2, for 2 columns.

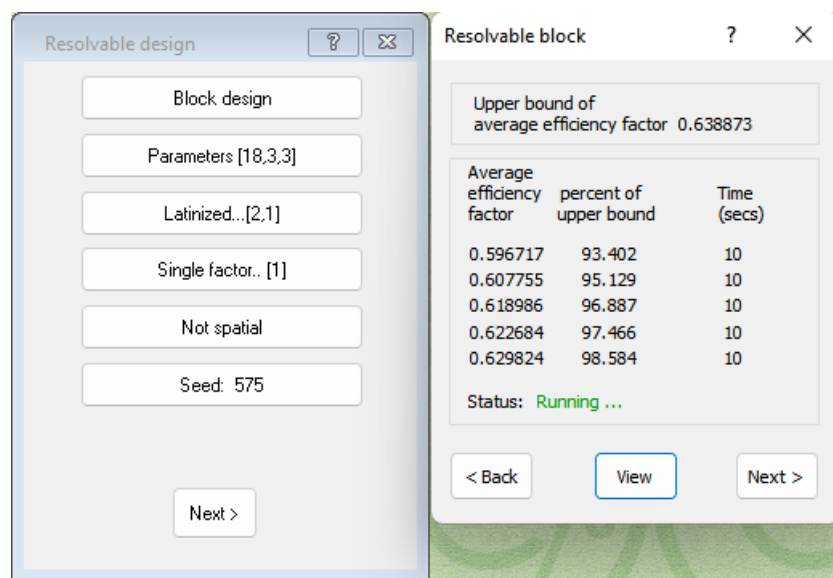
1. From the main menu click **Design** then select **Resolvable**.
2. Toggle the top button so that it displays **Block design** then click **Parameters** and enter the following values.



3. Click the **Not Latinized/Latinized** button and select *Latinized*.
4. Increase the *Number of contiguous blocks* to 2. This sets 't' to 2, and means that the iterative process tries as much as possible to equalize the number of times each treatment occurs in each pair of columns (not just within single columns).



5. Click **Next** then **Next** again to start the iterative process. For this design, even after 30 minutes, the Average efficiency factor is still only 98.5% of the estimated optimum:



6. It is unlikely that there would be much more improvement even after a few hours, so click **Next** to move to the *Randomization/Output* menu and click **Next** to generate the design.

```

Design properties
  Resolvable
  Block design
  Single factor

Design parameters
  Number of treatments           = 18
  Number of units/block         = 3
  Number of blocks               = 6
  Number of replicates          = 3

Latinized (t-latinized columns=2)

Random number seed for design generation = 575

Average efficiency factors (Upper bounds)
  Block      0.629824 (0.638873)

Concurrencies

      Concurrence      Block
      0                99
      1                54

Random number seed for design randomization = 430

Randomized design 1

+-----+
block   1   2   3   4   5   6
plot   +-----+
  1 |  11   1   7  15  10   9
  2 |  18   6   8   5  12   2
  3 |  16  13   3   4  14  17

  1 |  10   4  17  18   3   5
  2 |  15   8  12   1   6   7
  3 |   9  14  13   2  16  11

  1 |  17   2   9  11   4  15
  2 |   3   7  14   6  13   1
  3 |   5  12  16  10  18   8

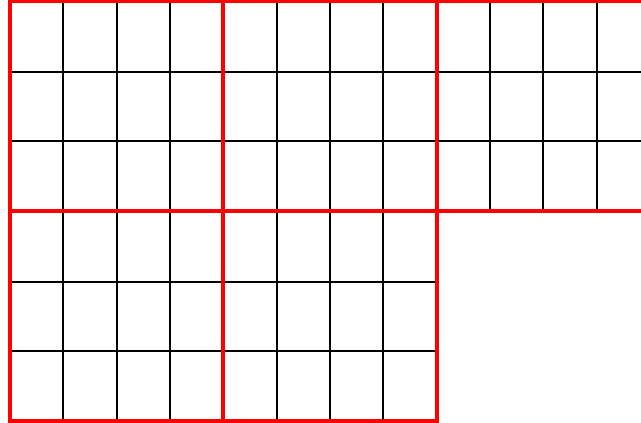
```

Note that the Latinization is summarised after the main design summary: the treatments were Latinized down the columns, with Latinizing for pairs of columns. In the final design you can see that there is a complete set of the 18 treatments in each lot of 3 rows by 6 columns, and also in the sets of 9 rows by 2 columns.

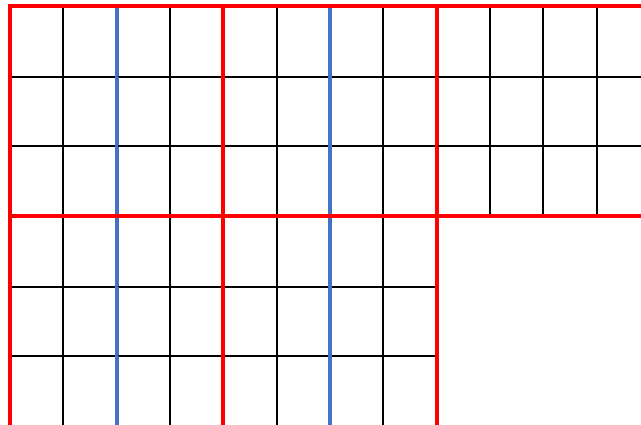
Latinizing row-column designs

This works similarly to Latinizing block designs, except that Latinizing can be done both in the column (top to bottom) and row (left to right) directions.

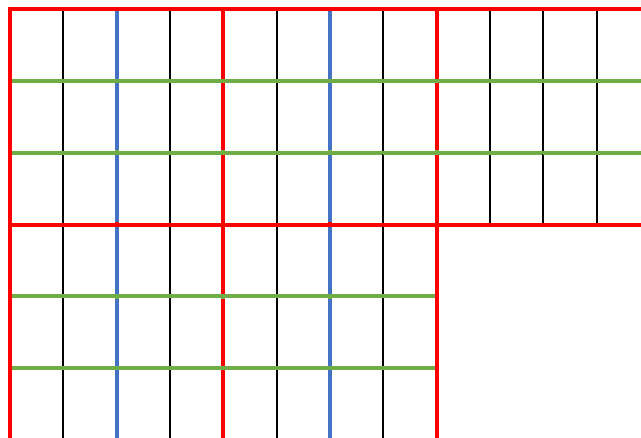
In this example there were 12 treatments ($v=12$), with each replicate laid out as 4 rows of 3 plots ($nr=4$; $nc=3$). There were 5 replicates of the treatments ($r=5$), with the replicates to be laid out as follows.



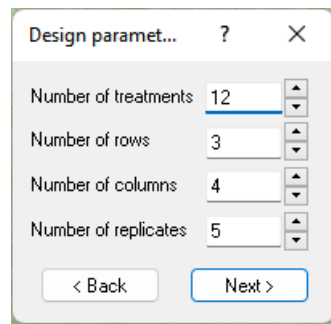
2-Latinizing the columns would create a doubly-resolvable design, with the main replicates and also replicates of 6 rows by 2 columns.



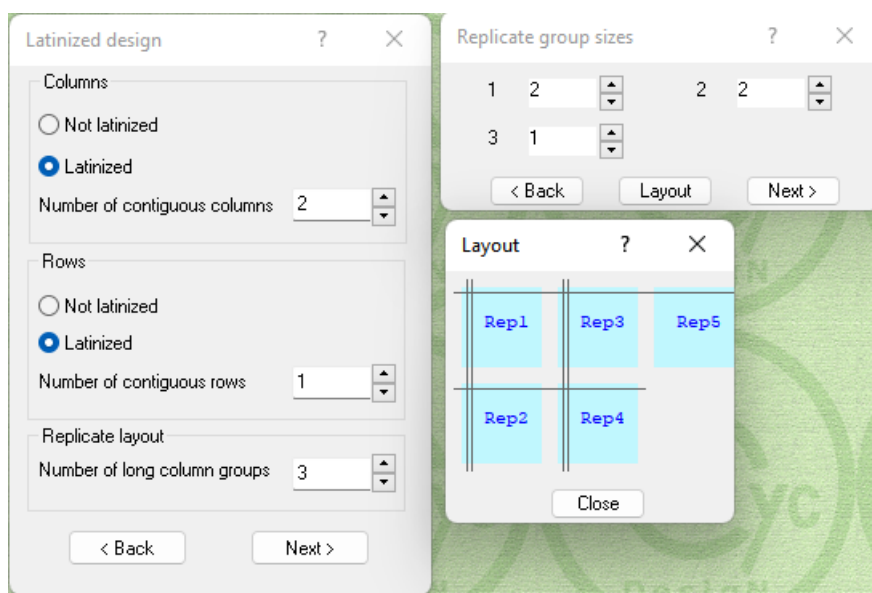
In addition, the rows can be Latinized across the whole design (3 lots of 1 row x 12 plots; 3 lots of 1 row x 8 plots):



1. To generate this design, from the **Design** menu select **Resolvable** then toggle the top button to display **Row-column**.
2. Set the parameters as follows and click **Next**.

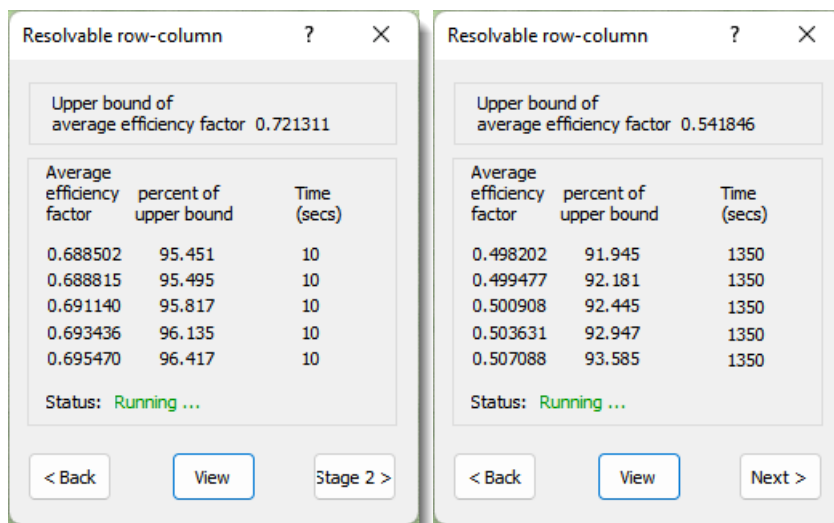


3. Click the **Not Latinized** button then select *Latinized* for both columns and rows. Change the *Number of contiguous columns* to 2 and *Number of long column groups* to 3 (there are 3 replicates going left to right).



4. Click **Next** and set the *Replicate group sizes* to 2, 2 and 1, then click **Layout** to check you have set things up correctly. The double vertical lines in the **Layout** indicate that the columns are 2-Latinized (Latinizing pairs of columns). The single horizontal line indicates that row-Latinizing is to be done row by row.

- Click **Close** then **Next**. Click **Next** in the main menu to start the design generation. If Two stage is used the first stage does not progress after half an hour so click **Stage 2 >**.



- Stage 2 also does not progress much, so click **Next** twice to generate the final design when you feel ready! (This is an example where choosing One stage could potentially reach a design with a higher Average efficiency.)

column	1	2	3	4	1	2	3	4	1	2	3	4
row 1	4	1	12	6	3	5	9	7	10	8	11	2
2	5	9	2	11	8	12	10	1	7	4	3	6
3	10	3	7	8	2	6	11	4	12	9	5	1
1	8	12	5	4	1	10	3	2				
2	6	2	9	10	11	7	12	5				
3	7	11	1	3	4	9	6	8				

Note that despite the Average efficiency not reaching even 95% of the upper bound, the main required features of the design have been satisfied: each treatment appears exactly once in each pair of columns (green rectangles), and similarly only once in each long row (orange rectangle).

Latinizing places constraints on the design, so should be used with care: with smaller numbers of treatments and/or replicates, Latinization, especially in both rows and columns, can produce systematic looking designs, or fail to achieve any reasonable level of efficiency.

The example above effectively has blocking in THREE dimensions. If replicates 1 and 2 are placed on a shelf in a controlled environment cabinet, replicates 3 and 4 on a second shelf and replicate 5 on a third shelf directly under replicates 1 and 3, the Latinizing allows for trends across shelves, left to right, and front to back to be adjusted for.

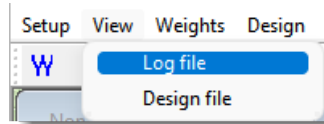
Explanation of design generation options and buttons in CycDesignN

This section describes the various options and settings that modify the design generation process.

The options on the main menu (**Setup**, **View**, **Weights**, etc) are explained in detail in the CycDesignN help, which you can open by clicking **Help**.

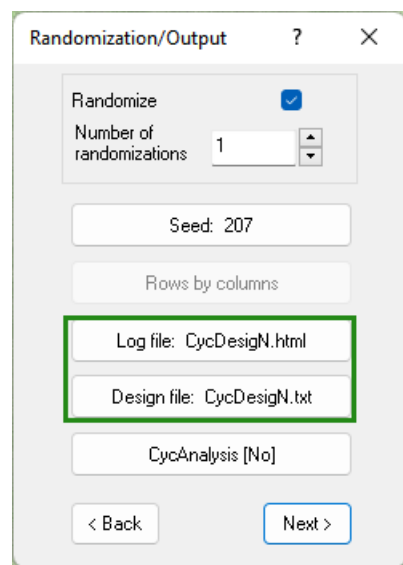
Where are my files stored?

To re-open a closed Log file or design file, from the menu click **View** then select **Log file** or **Design file**. By default, files are stored in C:\ProgramData\DATA.



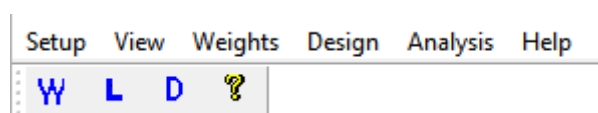
Changing file names

The names of the Log file and design file can be changed in the *Randomization/Output* menu. Clicking the **Log file** (or **Design file**) button opens up an Explorer window, where you can browse to change the folder as well as changing the name. The name (and location) used for the Log file CycDesignN is also used for the CycDesignN.csv file – only the file extension differs. In addition, the name CycDesignN is also used for a special .aux file, which contains the design information to allow the design to be used with CycAnalysis. For a full description of CycAnalysis please refer to the Help.



Changing the location of saved files

The three blue buttons immediately below the menu are used to navigate to your saved files. You can also save CycDesignN's output files to different locations using the **Log file** and **Design file** buttons on the *Randomization/Output* menu.



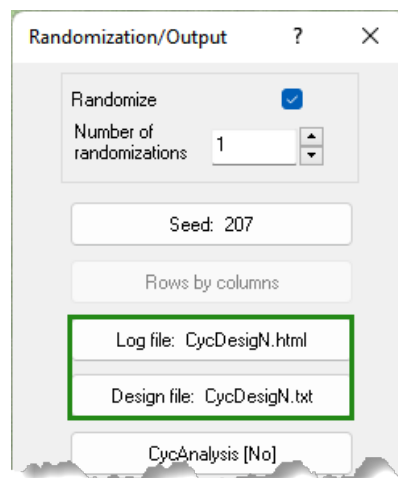
Working directory

W stands for *Working Directory* where your generated files are stored. By default the working directory is located in C:\ProgramData\DATA. To change the working directory click **W** then click **Browse** and navigate to a different location.

Log file

L stands for *Log file*, which is named CycDesigN.html by default. This is the file which opens automatically when the design generation process has finished. The Log file contains your completed design layout in graphic form along with the parameters, options and randomization seeds used to produce the design. You can use this as a record of how your design was constructed so that you can reproduce it exactly.

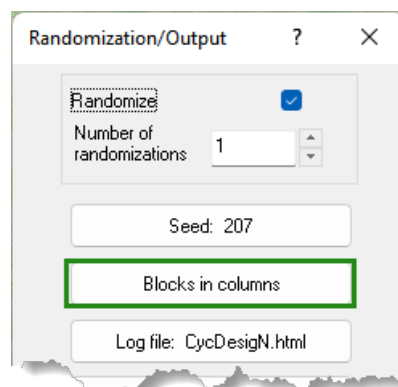
Warning: Every time you generate a new design this will overwrite CycDesigN.html. You can avoid overwriting the previous file by clicking **Log file** in the *Randomization/Output* menu and renaming the new file being generated, preferably with a name that describes the content/purpose.



Design file

D: stands for *Design file*, which is named CycDesigN.txt by default. This is a slightly less detailed version of the Log file that prints in black and white rather than colour. This text file makes it easy to copy and paste the layout into a spreadsheet or Word document – in fact this is why the original developer created the design file.

More importantly, you can use the design file to show the blocks of the design in either rows or in columns by toggling the button **Blocks in columns / Rows by columns**.



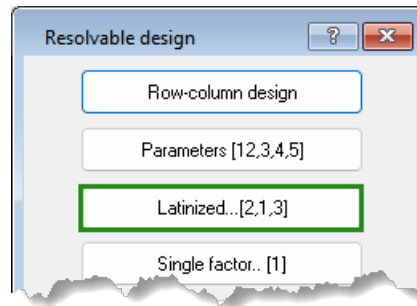
Warning: Every time you generate a new design this will overwrite CycDesigN.txt. You can avoid overwriting the previous file by clicking **Design file** in the *Randomization/Output* menu and renaming the new file being generated, preferably with a name that describes the content/purpose.

Help

? : Opens the CycDesignN Help.

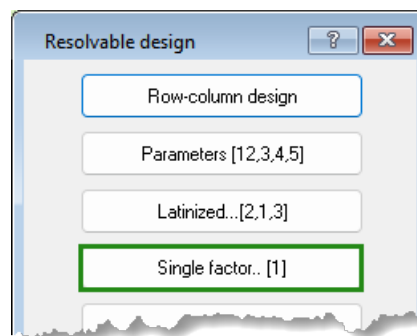
Latinizing

The CycDesignN Latinizing facility is one of the most useful capabilities of the program. For resolvable designs, it allows further blocking of plots, usually across the main blocking (or row or columns).



Factorial or nested treatments

For most block and row- column designs, the treatments can be given structure.

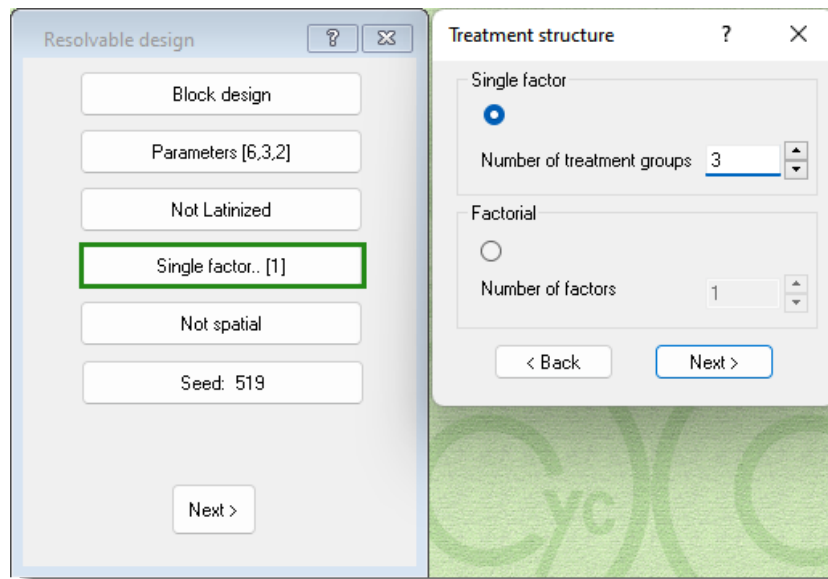


There are 2 sorts. These are illustrated as a modification of [Example 1](#), which has 6 treatments in all.

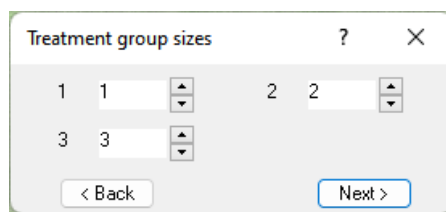
Single factor with the levels grouped and nested

There are 3 types of treatments (treatment groups): Treatment 1 is a standard variety. Treatments 2 and 3 are designated as 'group 1' varieties, and Treatments 4, 5, and 6 are 'group 2' varieties.

1. Set up a resolvable block design with parameters as shown below then click **Single factor**.
2. Set *Number of treatment groups* to 3 then click **Next**.



3. Change the numbers to match the groups sizes (1, 2 and 3) then click **Next**.



4. Click **Next** on subsequent menus to generate the design and open the Log file.

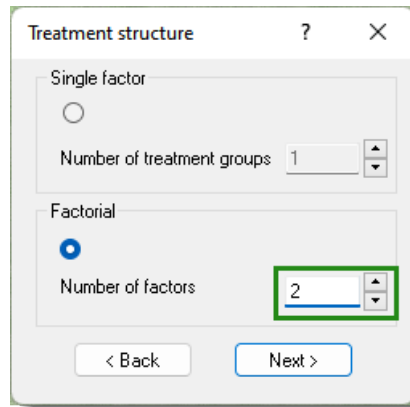
Instead of a single Treatment column, the .csv file will have a column for the factor groups (3 levels), and one for the treatment numbers, numbered across the groups. The Log file will indicate that the treatments are grouped.

```
Nested (Optimal)
Number of nested groups = 3
Group 1 Size 1 Treatments 1,2, ...
Group 2 Size 2 Treatments 2,3, ...
Group 3 Size 3 Treatments 4,5, ...
```

Factorial

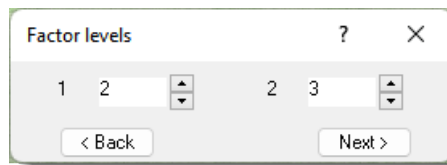
Factorial allows you to generate a set of treatments that comprise all combinations of the levels of these factors present in the design. For [Example 1](#) with 6 treatments, this can be modified to have a factorial set of factor A with 2 levels and factor B with 3 levels.

1. Set up a resolvable block design with parameters as shown below then click **Single factor**.
2. Select *Factorial* and set *Number of factors* to 2, then click **Next**.



The screenshot shows a dialog box titled "Treatment structure". It has two sections: "Single factor" and "Factorial". In the "Single factor" section, there is a radio button and a "Number of treatment groups" spinner set to 1. In the "Factorial" section, there is a selected radio button and a "Number of factors" spinner set to 2. At the bottom, there are "< Back" and "Next >" buttons.

3. Enter the number of levels for each factor as shown then click **Next**.



The screenshot shows a dialog box titled "Factor levels". It contains two spinner boxes. The first spinner is set to 2, and the second spinner is set to 3. At the bottom, there are "< Back" and "Next >" buttons.

4. Click **Next** on subsequent menus to generate the design and open the Log file.

Instead of a single Treatment column the `.csv` file will have a column for each of the factors A and B. The Log file will be modified similarly, with treatment pairs instead of single numbers e.g, (2, 1) means level 2 of factor A and level 1 for factor B. The `.csv` file will have a column for each of the two factors.

```
Factorial design average efficiency factors

      Effect      Ave Efficiency Factors      (Upper bounds)      (Weights)
      E1          0.857143          (0.888889)          ( 0.800)
      E2          1.000000          (1.000000)          ( 0.800)
      E12         0.461538

Random number seed for design randomization = 554

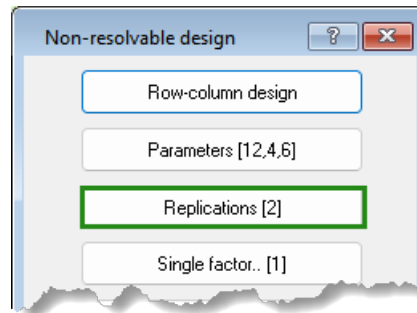
Randomized design 1

block  +-----+
      1  2
plot  +-----+
  1 | (2,1) (1,1)
  2 | (1,3) (2,3)
  3 | (1,2) (2,2)

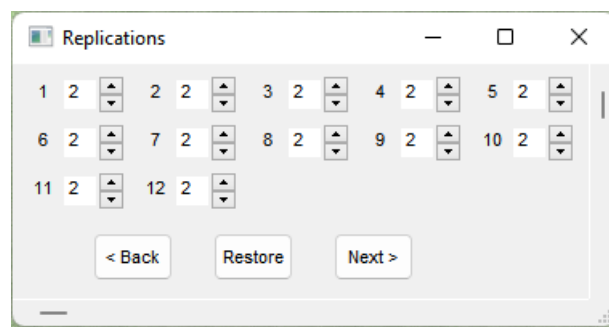
  1 | (1,3) (2,3)
  2 | (2,1) (1,1)
  3 | (2,2) (1,2)
```

Unequal replication

For a non-resolvable block or row-column design the **Replications** button allows differing numbers of replicates to be entered for the treatments.

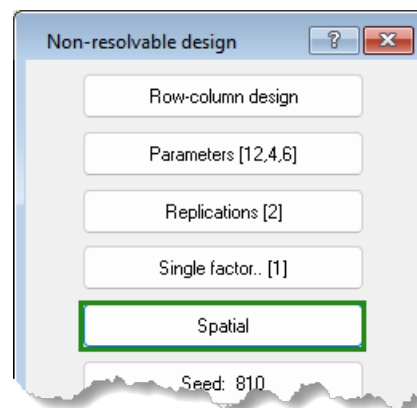


For [Example 4](#), clicking **Replications** opens the following menu, where the number of replicates can be changed, such that the total number of plots remains the same.



Spatial

When this button is enabled it will toggle on or off optimization of *Evenness*, *Neighbour balance* and *Treatment span*.



See Williams, E. & Piepho, H. 2019. Error variance bias in neighbour balance and evenness of distribution designs. *Australian & New Zealand Journal of Statistics* **61**(4), 466-473

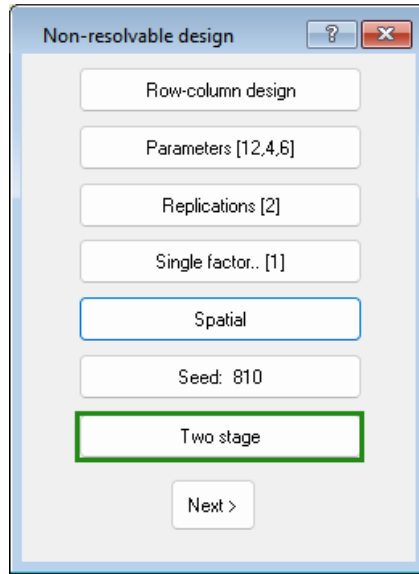
<https://onlinelibrary.wiley.com/doi/full/10.1111/anzs.12277>

Seeds

A *seed* is used to initialize an iterative or randomization process. By using a given seed, the iterative or randomization process will be completely replicated. This is useful if you want to generate a design more than once with exactly the same final layout. CycDesignN has 2 seeds: the first is used to start the iterative process. CycDesignN uses a second seed to start the randomization process.

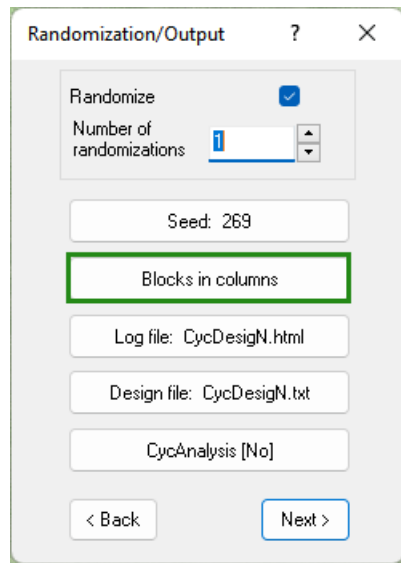
Two stage/One stage

This button is available for row-column designs. The default (**Two stage**) optimizes the columns first, and then the rows. If the button is clicked to give **One stage**, both dimensions are optimized at the same time. This can sometimes result in a design with a higher level of efficiency than that obtained with a Two stage optimisation, but often will be slower than a Two stage optimization.



Blocks in columns

When enabled, this button can sometimes be used to change the orientation of the trial layout as shown in the Log or Design file.

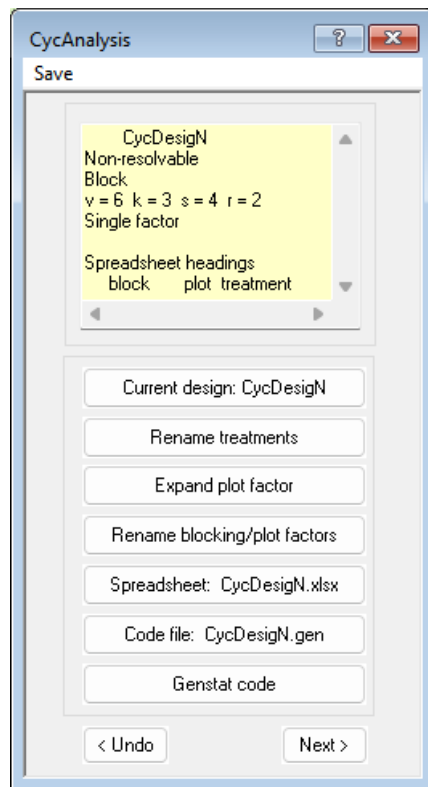


CycAnalysis

CycAnalysis allows some further output and modification of the generated design. When CycAnalysis is selected a `.xlsx` file is generated that contains column headings. For a full description of CycAnalysis please refer to the Help.

1. You can choose CycAnalysis at the Randomization step or click **Analysis | CycAnalysis** on the main menu to modify details of a design previously generated.
2. Select **Current design** to select an already generated `.aux` file to modify that design.

The other buttons change details of the design.



Rename treatments: Change the labels of the treatments from numbers to something else (e.g. A, B, C, etc.).

Expand plot factor: If you have several sub-units in a plot (e.g. a plot comprising 3 plants in a pot), this button can create a sub-factor such as 'plant in pot' and expand the other factors to match.

Rename blocking/plot factors: Rename the generated factors.

Spreadsheet: Change the name of the Excel spreadsheet (and the `.csv` file) which will contain the factor columns. This spreadsheet (and the `.csv` file) may contain a column of random numbers if it is feasible to generate a Code file.

Code file: Genstat (or SAS) code: Change the name of a file to hold commands to analyse data from the file. The Code will be Genstat if the next button displays **Genstat code** or SAS if that button displays **SAS code**.