Designing Computer Experiments: Rotated Factorial Designs

Scott D. Beattie*

The Pennsylvania State University, 331-B Thomas Building, University Park, PA 16802

Dennis K. J. Lin†

The Pennsylvania State University, 314 Beam Building, University Park, PA 16802

Max D. Morris‡

Oak Ridge National Laboratory, Mathematical Sciences Section, CSM Division, Oak Ridge, TN 37831-6367

Abstract

Computer models can describe complicated physical phenomena, such as performance characteristics of integrated circuits. However, to use these models for scientific investigation, their generally long running times and mostly deterministic nature require a special designed experiment. Standard factorial designs are inadequate; in the absence of one or more main effects, their replication cannot be used to estimate error but instead produces redundancy. A number of alternative designs have been proposed, but many can be burdensome computationally. This paper presents a new class of designs developed from the rotation of a two-dimensional factorial design in the plane. These rotated factorial designs are very easy to construct and preserve many of the attractive properties of standard factorial designs: they have equally-spaced projections to univariate dimensions and uncorrelated regression effect estimates (orthogonality). They also rate comparably to maximin Latin hypercube designs by the minimum interpoint distance criterion used in the latter's construction.

Key Words

effect correlation, Latin hypercube, maximin distance, minimum interpoint distance

*Mr. Beattie is a Ph.D. candidate in the Department of Statistics.

†Dr. Lin is an Associate Professor in the Department of Management Science and Information Systems and in the Department of Statistics.

‡Dr. Morris is a Research Staff Member.

1 Introduction

Computer models are often used to describe complicated physical phenomena encountered in science and engineering, such as weather patterns or performance characteristics of integrated circuits. These physical phenomena are often governed by a set of equations, including linear and nonlinear equations and ordinary and partial differential equations, too difficult to be solved simultaneously by any person, but can be by the computer modeling program. These programs, due to the number and complexity of the equations, often have long running times which make them difficult to use for comprehensive scientific investigation.

The SOLA-PTS algorithm described in Daly and Torrey (1984), for example, has been developed at the Los Alamos National Laboratory for modeling the rapid cooling of a nuclear reactor wall as a result of cold water injected into the reactor's downcomer for containment during a nuclear accident. The authors' three-pronged goal is to study the response of the reactor, to study the turbulent mixture of the cold water and the warm fluid already in the downcomer, and to predict the onset and growth of cracks in the reactor wall as a result of the rapid cooling. This algorithm simultaneously solves eight partial differential equations with eight inputs and takes approximately 90 minutes on a Cray supercomputer to run. This algorithm is typical of computer models needing designed experiments. It solves a large number of differential equations, is very computationally expensive in running time, and has a "black box" quality – one does not know in advance which factors have large effects and one would like to examine the response over a wide range of input combinations. A design with properties like separability of effects, equally-spaced projections to each dimension, and high spatial dispersion seems appropriate for this setting.

Of the many goals in this setting, one is to build an approximating program which, although not as precise as the computer model, would run fast enough to study the phenomenon in detail. Construction of an adequate approximating function (or program) to the computer model requires the selection of design points (a designed experiment) at which to approximate. These computer experiments, because the computer models are mostly deterministic, require special designs. In physical experiments, standard factorial designs work well. If certain factors have no effect on the response and are taken out of the approximation function (linear model), then the replicated design points in the reduced design space can be used to estimate the random error present in the system. However, with computer experiments, there is no random error – only lack of fit. Standard factorial designs are inadequate; in the absence of certain main effects, replication cannot be used to estimate this error, but instead produces redundancy. That is, they have poor projection properties to lower dimensions.

A number of alternative designs have been proposed and will be discussed in the next section. The rest of this paper presents a new and simple strategy for designs for computer experiments, developed from the rotation of the two-factor standard factorial design in the plane. For simplicity, this paper focuses exclusively on the two-dimensional case. Because these designs are a new idea, we have chosen to limit the scope so that we may present the design strategy precisely and clearly. Unfortunately, the extension of these designs to higher-dimensional design spaces involves many careful steps and we do not want the basic idea here to get lost in a sea of technicalities or to overwhelm our audience. Preliminary results for the extension are promising and we defer them to another presentation in the near future. The following sections develop the rationale for these designs and compare them with the other alternative designs by various criteria. Finally, the appendix provides an S-Plus function to implement these new designs.

2 Previous Work and Comparison Criteria

Koehler and Owen (1996) outlined the many goals in computer experiments: optimization of the response over the inputs, visualization of the response over the input space, approximation of the response over a subregion, integration of the response over a subregion, and determination of the factors which are most important to the response. The selection of an appropriate experimental design may depend on the goal at hand. To this end, they give an excellent overview of the many available strategies which they categorize as either "frequentist designs" or "bayesian designs."

The frequentist designs are characterized by geometric properties and are often highly structured. They may be completely determined or created by sampling from a completely determined structure. These designs include standard (full or fractional) factorial designs, Latin hypercube designs, and many other traditional designs used in physical experiments. Although standard factorial designs have poor projection properties, they and other frequentist designs have many properties that are useful for computer experiments. Their design simplicity, especially the equally-spaced design points and projections, makes the goals of visualization and integration easy and of approximation and determination straightforward. Their orthogonality aids in the goals of integration and approximation while making determination straightforward. Finally, they ensure that all regions of the design space are represented, aiding in optimization.

The bayesian designs are created by optimizing some quantity when a prior distribution is put over the design space. Creation of these designs is computer-intensive as they must be obtained by some method of computer search, like kriging. These designs include those of Sacks, Schiller, and Welch (1989) and Sacks, Welch, Mitchell, and Wynn (1989) that minimize the integrated mean square error of prediction when the prediction errors are taken as a realization of a spatial stochastic process.

Johnson, Moore, and Ylvisaker (1990) proposed similar designs that minimize the determinant of the correlation matrix of the responses when these responses are taken as a realization of a spatial stochastic process. That is, they attempt to minimize the correlations between observations. They proved that the designs that attain this are maximin distance designs – those that achieve the largest distance possible between the two closest design points. They call a design D^* a maximin distance design if

$$\min_{x_1,x_2 \in D^*} d(x_1, x_2) = \max_{D} \min_{x_1,x_2 \in D} d(x_1, x_2),$$

where d is a suitably chosen distance measure and $\min_{x_1,x_2 \in D} d(x_1, x_2)$ is the minimum interpoint distance of design D.

An unfortunate drawback to maximin distance designs is that they are not always good designs for computer experiments. For example, the 4 point maximin distance design in 2 dimensions is exactly the 2^2 factorial design, which has poor projection properties, because the maximin criteria says to move the design points as far apart as possible within the square design space – to the corners. In fact the maximin distance design with $n = p^2$ design points, where p is an integer, is always the standard factorial design.

Other problems with maximin distance designs and other bayesian designs include their computer-intensive nature and often non-geometrical appearance. It has been argued that these designs may be better for the goal of approximation (though this is not clear), but they make the goals of visualization and determination more difficult and may not be suitable for optimization. Easterling (1989), in his comment to Sacks, Welch, Mitchell, and Wynn (1989), describes these designs as "ugly" and argues that the attractive properties of factorial designs should not be discarded so easily.

Morris and Mitchell (1992) attempted to bridge the gap between frequentist designs and bayesian designs by proposing a type of maximin distance design derived from the Latin hypercube. First, a Latin hypercube with n design points is generated by randomly permuting the integers $\{1, \ldots, n\}$ for each factor. Next, the authors iteratively permute this hypercube to improve its minimum interpoint distance until, hopefully, the hypercube with largest minimum interpoint distance is attained. A strength of this approach is that the resulting design is a Latin hypercube so that it projects into n equally-spaced points in each dimension. Drawbacks to this methodology include the need to use computer search to find these designs and the possibility that local, not global, optimal designs will be found by the algorithm.

3 Rotated Factorial Designs in Two Dimensions

As Easterling (1989) points out, standard factorial designs have many attractive properties for physical experiments: balance, symmetry, orthogonality, collapsibility, equally-spaced projections to each dimension, and straightforward measurability of main effects. Factorial designs are simple to construct, yet powerful – in a word, elegant. Although the collapsibility of factorial designs is not an asset in the setting of computer experiments (as explained in the introduction), the other properties of factorial designs are worth preserving. Our goal is to construct a class of designs for computer experiments that preserve many of these properties.

Our strategy is to modify the standard factorial design to make it more appropriate for computer experiments. We suggest rotating the standard factorial design to reduce the number of redundant projections to lower dimensions. In fact, it is possible to rotate in such a way that the design points yield unique, equally-spaced projections to each dimension. To see how this is done in general, first consider the standard 3^2 factorial design, represented by the 3×3 square of points in Figure 1, and how it can be rotated to yield equally-spaced projections. The key to finding all such rotations is in the relationship between points A-D. We focus on nontrivial angles between 0 and 45 degrees clockwise due to the symmetry of the rotation problem.

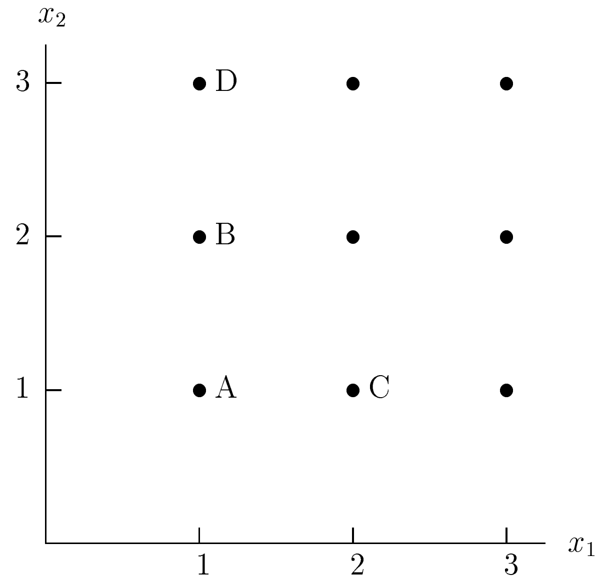

Figure 1: Standard 3^2 factorial design before rotation

The matrix equation to rotate a set of points clockwise by an angle w about the origin is

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \times \begin{bmatrix} \cos(w) & -\sin(w) \\ \sin(w) & \cos(w) \end{bmatrix},$$

so that if (x_1, x_2) are the coordinates of a design point in the standard factorial design, then the rotation moves the point to $(x_1 \cos(w) + x_2 \sin(w), -x_1 \sin(w) + x_2 \cos(w))$.

Notice first that as the points are rotated clockwise about the origin that point A will have the smallest x_1-coordinate for any angle of rotation between 0 and 45 degrees. (A 45 degree rotation will place point A directly on the x_1-axis and A is the closest point to the origin.) Also notice that the x_1-projections of points with the same initial x_1-coordinate (like A, B, and D) will be equally spaced, by $\sin(w)$, regardless of the rotation angle. Likewise, the x_1-projections of points with the same initial x_2-coordinate (like A and C) will be equally spaced, by $\cos(w)$, regardless of the rotation angle. It suffices to find all angles that make the x_1-projections of points A-D equally spaced. For the x_1-coordinates of points A-D, see the table below.

point	x_1-coordinate
A	$\cos(w) + \sin(w)$
B	$\cos(w) + 2\sin(w)$
C	$2\cos(w) + \sin(w)$
D	$\cos(w) + 3\sin(w)$

Between 0 and 45 degrees, $\sin(w) \leq \cos(w)$, so the point with the next smallest x_1-coordinate will always be B (although C will tie B when $w = 45$ degrees). Therefore, the distance between the smallest two x_1-projections will always be $\sin(w)$. In order to achieve equally-spaced x_1-projections, the distance between all x_1-projections must equal $\sin(w)$. We've already seen that this is true when $w = 45$ degrees (equivalently, $\tan^{-1}(1)$) and both C and B have the second smallest x_1-coordinate.

Another possibility is that C will have the third smallest x_1-coordinate, and that the "x_1-distance" between B and C will be $\sin(w)$. However, the "x_1-distance" between B and D is always $\sin(w)$. In this case, C and D will have the same x_1-coordinate, hence

$$\cos(w) = 2\sin(w) \Longrightarrow w = \tan^{-1}(1/2).$$

Continuing in this manner, consider the case where C has the fourth smallest x_1-coordinate – after A, B, and D – and the "x_1-distance" between D and C is $\sin(w)$. Then

$$\cos(w) - 2\sin(w) = \sin(w) \Longrightarrow \cos(w) = 3\sin(w) \Longrightarrow w = \tan^{-1}(1/3).$$

It is not possible for C to have the fifth smallest x_1-coordinate, so these three rotations are the only ones (again, among nontrivial angles between 0 and 45 degrees) that yield equally-spaced x_1-projections from the 3^2 design. It can be easily verified that these three rotations also yield equally-spaced x_2-projections.

Figure 2 displays these three rotations of the standard 3^2 factorial design, shown in open circles, and the designs that result from them, shown in solid circles. For reference, boxes are drawn around the rotated designs to identify the design spaces. Along each axis, we have provided dot plots of the projections from which it is plain to see the equally-spaced property. Figure 3 is a similar display of the four rotations of a standard 4^2 factorial design that yield equally-spaced projections.

Following the argument above, a general result for factorial designs can be stated as in Theorem 1.

Theorem 1 *For nontrivial rotations between 0 and 45 degrees, a rotated standard p^2 factorial design will produce equally-spaced projections to each dimension if and only if the rotation angle is $\tan^{-1}(1/k)$ where $k \in \{1, \ldots, p\}$.*

Among the several rotated standard p^2 factorial designs which have equally-spaced projections, those obtained from rotation angles of $\tan^{-1}(1)$ are attractive due to their symmetry (see Figures 2(a) and 3(a)). However, these designs contain many redundant projections (and

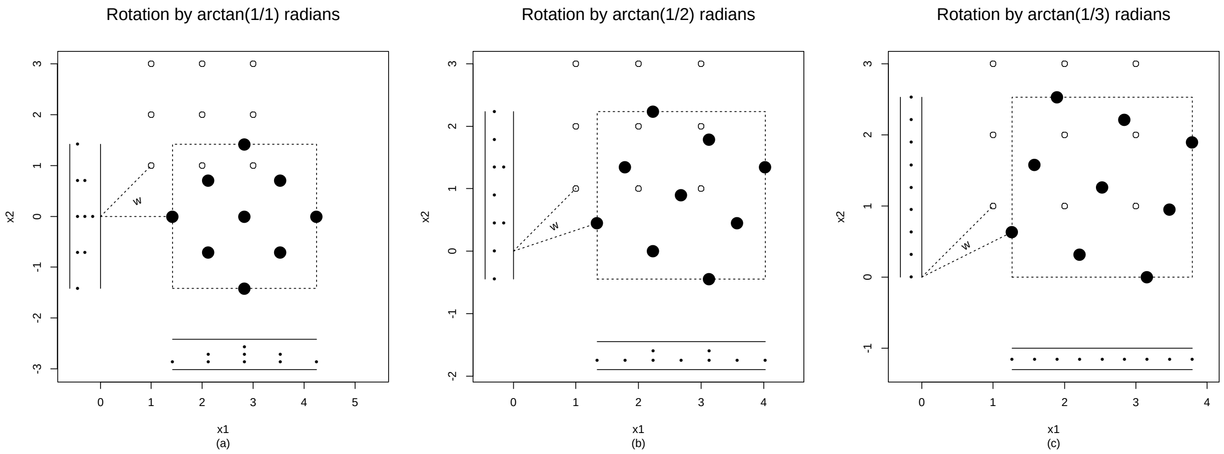

Figure 2: Three rotations of a standard 3^2 factorial design:

(a) $w = \tan^{-1}(1)$, (b) $w = \tan^{-1}(1/2)$, (c) $w = \tan^{-1}(1/3)$

imbalance). On the other hand, the designs obtained from rotation angles of $\tan^{-1}(1/p)$, despite their slight asymmetry, have no redundant projections (hence balance, see Figures 2(c) and 3(d)). If one of the factors has no effect on the response, the latter designs yield p^2 equally-spaced points in the reduced design space for which to measure the effect of the other factor. Therefore, we define a p^2 *point rotated full factorial design* to be a rotated standard p^2 factorial design with unique, equally-spaced projections to each dimension.

Figure 4 shows the 4, 9, and 16 point maximin Latin hypercube designs published by Morris and Mitchell (1992). For comparison, the familiar projection dot plots have been included on the graphs. The 4 and 9 point designs are identical to the corresponding 4 and 9 point rotated full factorial designs. The two 16 point designs, although different, have equal minimum interpoint distances, $\min_{x_1,x_2 \in D} d(x_1, x_2)$, of 0.2749 using Euclidean distance for d. In fact, every p^2 point rotated full factorial design is equivalent to the p^2 point maximin Latin hypercube design by the minimum interpoint distance criterion. However, rotated full factorial designs are easier to develop – no computer search is required.

4 Subset Designs

Rotated full factorial designs can be easily modified to accomodate many design sizes other than p^2 for p an integer. After rotating the standard factorial design, remove the four most extreme points – two for each factor – to get a new design. In other words, if X_1 and X_2 are the two factors, then to get this new design, remove the points with the smallest X_1 value, largest X_1 value, smallest X_2 value, and largest X_2 value. This process can be repeated to get any design with number of points equal to $p^2 - 4j$, where j is the number of times this deletion process is performed and $j \in \{0, 1, \ldots, \max(p-2, 0)\}$.

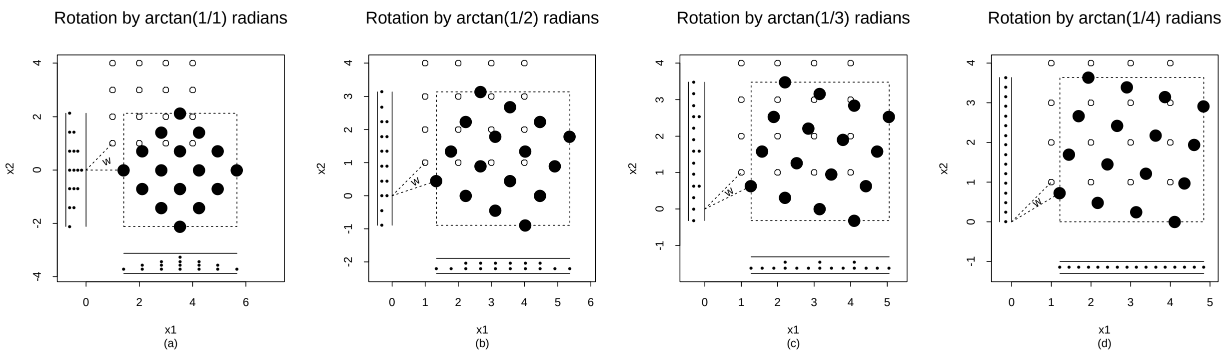

Figure 3: Four rotations of a standard 4^2 factorial design:

(a) $w = \tan^{-1}(1)$, (b) $w = \tan^{-1}(1/2)$, (c) $w = \tan^{-1}(1/3)$, (d) $w = \tan^{-1}(1/4)$

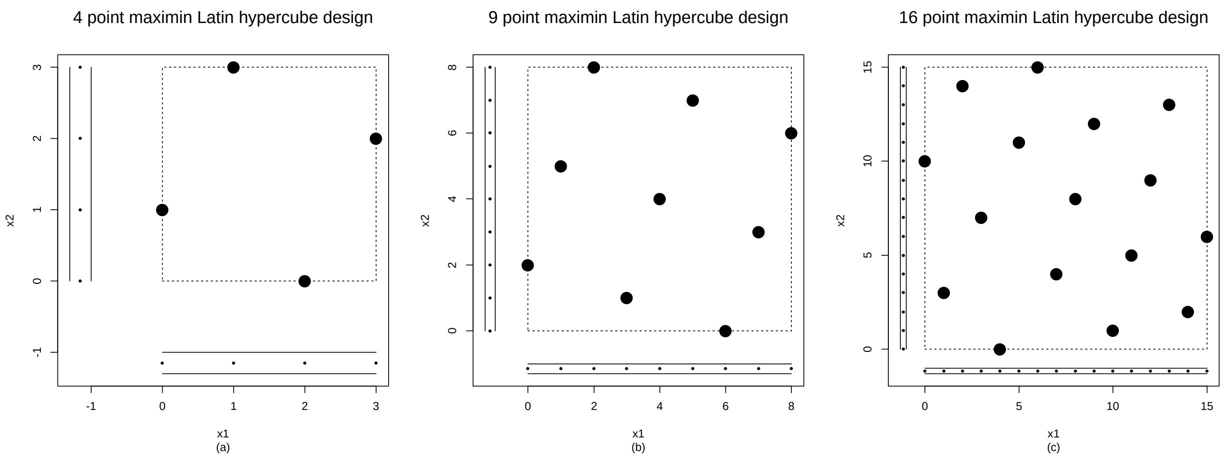

Figure 4: Morris' and Mitchell's 4, 9, and 16 point maximin Latin hypercube designs

Rotated full factorial designs were developed so that the design points created unique, equally-spaced projections to each dimension. When some of these points are removed through this deletion process, the resulting design will no longer have the equally-spaced projection property, although all design points will still project uniquely to each dimension. We will refer to designs created by applying the deletion process to a rotated full factorial design as *Type U rotated factorial designs*, where the letter U emphasizes that these designs have *unique* projections. Since it is possible to delete 0 points in the deletion process, all rotated full factorial designs are trivially Type U designs. Figure 5 shows the 12 point Type U rotated factorial design that is created by removing the four most extreme design points of the 16 point rotated full factorial design of Figure 3(d). Note that the design points project uniquely but that the projections are not equally spaced.

When points are removed from rotated full factorial designs through the deletion process, the new design can be given equally-spaced projections by adjusting the angle of rotation to $\tan^{-1}(1/k)$ for the largest $k \in \{p-2, p-1, p\}$ that makes the projections equally spaced,

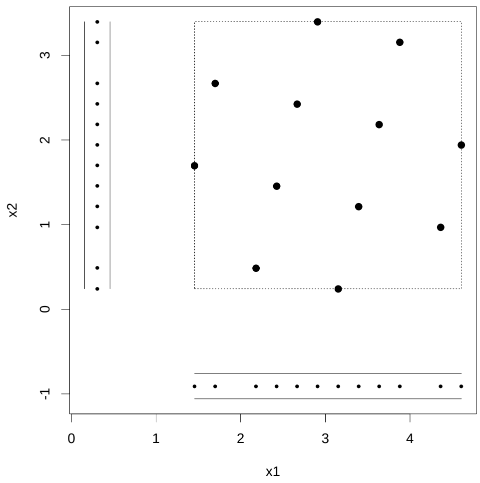

Figure 5: 12 point ($4^2 - 4$) Type U rotated factorial design

although this may have the simultaneous effect of creating redundant projections. (Picking the largest k will minimize the number of redundant projections.) We will refer to designs created by modifying the rotation angle of a Type U design to yield the greatest number of unique, equally-spaced projections as *Type E rotated factorial designs*, where the letter E emphasizes that these designs have *equally-spaced* projections. Since rotated full factorial designs are Type U designs with equally-spaced projections, they are also trivially Type E designs. Figure 6 shows the 12 point Type E rotated factorial design which has been given equally-spaced projections by adjusting the rotation angle to $\tan^{-1}(1/3)$ (see Figure 3(c)). Note that each dimension contains one redundant projection.

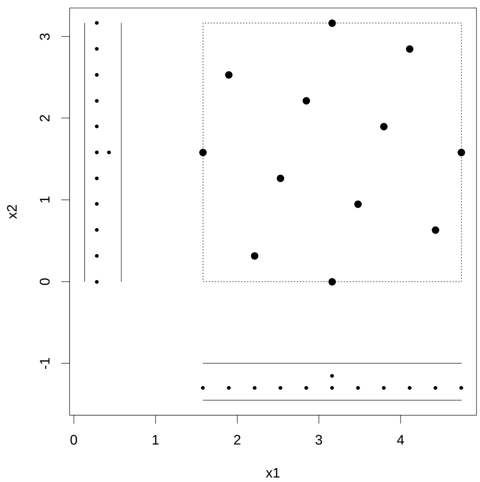

Figure 6: 12 point ($4^2 - 4$) Type E rotated factorial design

Figure 7 contains the 8 point Type U and Type E rotated factorial designs that result from removing the four most extreme design points from the 12 point Type U rotated factorial design. To get equally-spaced projections, the rotation angle of the Type E design is adjusted to $\tan^{-1}(1/2)$.

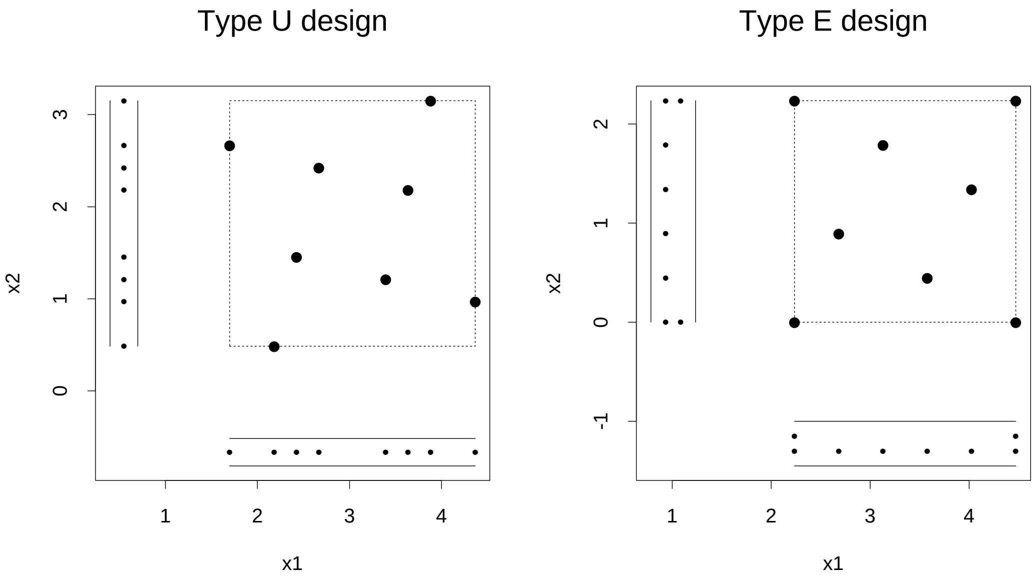

Figure 7: 8 point $(4^2 - 8)$ Type U and Type E rotated factorial designs

We leave it to the reader to choose between Type U and Type E designs in practice. Our preference is for Type E designs because of the equally-spaced projections. Others may choose Type U designs because of the unique projections and inherent balance. Still others may make their choice by some other criterion such as minimum interpoint distance.

5 Minimum Interpoint Distance Comparisons

Morris and Mitchell (1992) provided the design points for their maximin Latin hypercube designs for experiments with $n = 3, \ldots, 20$ points. Table 1 presents the minimum interpoint distances $(\min_{x_1, x_2 \in D} d(x_1, x_2))$, calculated by scaling the designs to the unit square $[0, 1]^2$ and using Euclidean distance, for these designs and for the rotated factorial designs. The asterisks indicate that no rotated factorial designs can be constructed as defined. Johnson, Moore, and Ylvisaker (1990) gave ranges for the minimum interpoint distances of maximin distance designs. These are listed merely as a reference for the other designs; no direct comparison will be made since maximin distance designs aren't necessarily appropriate for computer experiments. A few maximin distance designs were published in Johnson, Moore, and Ylvisaker (1990) and in Koehler and Owen (1996) and the exact minimum interpoint distances are listed for those designs.

In certain cases, the minimum interpoint distances of maximin Latin hypercube and rotated factorial designs are equal – most notably when there are p^2 design points, but also when

Table 1: Comparison of Minimum Interpoint Distances

Number of Points	Maximin Distance Design	Maximin Latin Hypercube	Rotated Factorial Design Type U	Type E
3	1.0000-1.4142	.7071	$*$	$*$
4	1.0000	.7454	.7454	.7454
5	.7071	.5590	.5270	.5590
6	.6009	.4472	$*$	$*$
7	.5314	.4714	$*$	$*$
8	.5000-1.0000	.4041	.3748	.4472
9	.5000	.3953	.3953	.3953
10	.3333-.5000	.3514	$*$	$*$
11	.3333-.5000	.3162	$*$	$*$
12	.3333-.5000	.3278	.3172	.3162
13	.3333-.5000	.3005	.2833	.3162
14	.3333-.5000	.3172	$*$	$*$
15	.3333-.5000	.2945	$*$	$*$
16	.3333	.2749	.2749	.2749
17	.2500-.3333	.2652	.2550	.2577
18	.2500-.3333	.2496	$*$	$*$
19	.2500-.3333	.2357	$*$	$*$
20	.2500-.3333	.2233	.2253	.2425

$*$ No rotated factorial design can be constructed as defined.

$n = 5$. In a few instances ($n = 8, 19, 20$) the minimum interpoint distances are better for rotated factorial designs, while maximin Latin hypercube designs are superior in the other cases ($n = 12, 17$). Recall that maximin Latin hypercube designs are constructed to have large minimum interpoint distances while preserving unique, equally-spaced projections, while rotated factorial designs are constructed with only unique, equally-spaced projections and design simplicity in mind. However, the gains in minimum interpoint distance by using maximin Latin hypercube designs, despite the significant increase in computer effort, are never very large when compared alongside maximin distance designs, the ideal according to minimum interpoint distance.

6 Further Comparisons – Estimator Dependency

The criterion for design comparison depends to a large extent on the final method of analysis. Although there have been a number of new methods proposed recently, let us assume that the research scientist performing the computer experiment is most familiar with linear regression. In this instance, a very desirable property for designs is for the regression effect estimates to be uncorrelated. Table 2 presents the correlations between regression effect estimates for the maximin Latin hypercube and rotated factorial designs. Only a few maximin Latin hypercube designs have uncorrelated regression effects, but all rotated factorial designs have this property, stated formally in Theorem 2 below. This is a great advantage for using rotated factorial designs.

Table 2: Correlation between regression effect estimates

Number of Points	Maximin Latin Hypercube	Rotated Factorial Design	
		Type U	Type E
3	-.5000	*	*
4	0	0	0
5	0	0	0
6	-.0286	*	*
7	-.1429	*	*
8	-.1429	0	0
9	0	0	0
10	-.2000	*	*
11	-.0091	*	*
12	0	0	0
13	.2143	0	0
14	.2088	*	*
15	.0143	*	*
16	.1265	0	0
17	.0588	0	0
18	.0588	*	*
19	-.1263	*	*
20	.0617	0	0

* No rotated factorial design can be constructed as defined.

Theorem 2 *Any rotated factorial design has uncorrelated regression effect estimates.*

Proof: See the appendix.

7 Concluding Remarks

The available design sizes for rotated factorial designs, given in Tables 1 and 2, are most likely sufficient in practice. However, further modification allows us to fill in many of the missing entries, if so desired. Treating the range of each factor asymmetrically, the aforementioned deletion process allows for the removal of an additional two design points. For example, one might delete the point with the largest X_1 value and the point with the smallest X_2

value, or with the largest X_1 value and the largest X_2 value. We will refer to designs created by removing two (mod 4) design points in the deletion process as *modified rotated factorial designs*. As before, this definition allows for both Type U and Type E modified designs. Using this modification, all but a few design sizes are available using rotated factorial designs.

Table 3 updates the minimum interpoint distances of Table 1 and the correlations between regression effect estimates of Table 2 to include modified rotated factorial designs. Again, as in Table 1, roughly one-third of the maximin Latin hypercube designs are better than the corresponding rotated factorial designs ($n = 7, 12, 14, 15, 17$), while in the other cases rotated factorial designs are superior ($n = 8, 13, 19, 20$) or just as good ($n = 4, 5, 9, 10, 16$). Although the modified designs do not have uncorrelated regression effect estimates (Theorem 2 applies only to unmodified designs), these correlations are remarkably small and superior – usually by a large amount – to maximin Latin hypercube designs. Certainly the minimal gains in minimum interpoint distance are negated by the significantly higher estimator dependencies of the maximin Latin hypercube designs and the computational complexity involved in constructing them.

Table 3: Further Comparisons of Rotated Factorial Designs

	Minimum Interpoint Distance				Effect Correlation		
No. of Pts.	Maximin Distance Design	Maximin Latin Hypercube	Rotated Factorial Design Type U	Rotated Factorial Design Type E	Maximin Latin Hypercube	Rotated Factorial Design Type U	Rotated Factorial Design Type E
3	1.0000-1.4142	.7071	*	*	-.5000	*	*
4	1.0000	.7454	.7454	.7454	0	0	0
5	.7071	.5590	.5270	.5590	0	0	0
6	.6009	.4472	*	*	-.0286	*	*
7	.5314	.4714	.4518	.4472	-.1429	.0462	.0616
8	.5000-1.0000	.4041	.3748	.4472	-.1429	0	0
9	.5000	.3953	.3953	.3953	0	0	0
10	.3333-.5000	.3514	.3436	.3514	-.2000	.0299	.0303
11	.3333-.5000	.3162	*	*	-.0091	*	*
12	.3333-.5000	.3278	.3172	.3162	0	0	0
13	.3333-.5000	.3005	.2833	.3162	.2143	0	0
14	.3333-.5000	.3172	.2945	.2875	.2088	.0100	.0127
15	.3333-.5000	.2945	.2684	.2875	.0143	.0125	.0108
16	.3333	.2749	.2749	.2749	.1265	0	0
17	.2500-.3333	.2652	.2550	.2577	.0588	0	0
18	.2500-.3333	.2496	*	*	.0588	*	*
19	.2500-.3333	.2357	.2428	.2425	-.1263	.0079	.0083
20	.2500-.3333	.2233	.2253	.2425	.0617	0	0

* No rotated factorial design can be constructed.

This paper has presented a new class of experimental designs for computer experiments. These rotated factorial designs are developed from the standard factorial design and, due to their geometrical elegance, share many of the qualities that make the latter so attractive for physical experiments. Rotated factorial designs are extremely simple to construct – even by hand – in contrast to the computer intensive nature of most of the designs introduced recently. The tradeoff for this design simplicity appears to be negligible, perhaps even nonexistant, by important design criteria.

An extension of rotated factorial designs to higher-dimensional design spaces is currently under investigation. Preliminary results are promising and the final results should be available in the near future.

The appendices provide some useful material: a proof to Theorem 2, a step-by-step sample construction of a rotated factorial design, and an S-Plus function that creates rotated factorial designs so that it is not necessary to construct them by hand in practice. Detailed instructions for using the S-Plus material are included.

References

Daly, B. J. and M. D. Torrey (1984). SOLA-PTS: A transient, three-dimensional algorithm for fluid-thermal mixing and wall heat transfer in complex geometries. Technical Report LA-10132-MS, Los Alamos National Laboratory.

Easterling, R. G. (1989). Comment: Design and analysis of computer experiments. *Statistical Science 4*(4), 425–427.

Johnson, M. E., L. M. Moore, and D. Ylvisaker (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference 26*, 131–148.

Koehler, J. R. and A. B. Owen (1996). Computer experiments. In S. Ghosh and C. R. Rao (Eds.), *Handbook of Statistics*, Volume 13, Chapter 9, pp. 261–308. Elsevier Science B. V.

Morris, M. D. and T. J. Mitchell (1992). Exploratory designs for computational experiments. Technical Report TM-12045, Oak Ridge National Laboratory.

Sacks, J., S. B. Schiller, and W. J. Welch (1989, February). Designs for computer experiments. *Technometrics 31*(1), 41–47.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments. *Statistical Science 4*(4), 409–423.

Appendix

A Proof of Theorem 2

Theorem 2 *Any unmodified rotated factorial design has uncorrelated regression effect estimates.*

Proof: First, we show that any rotated full factorial design has uncorrelated regression effect estimates. Begin with a p^2 full factorial design:

$$\begin{bmatrix} 1 & 1 \\ \vdots & \vdots \\ 1 & p \\ 2 & 1 \\ \vdots & \vdots \\ 2 & p \\ \vdots & \vdots \\ p & 1 \\ \vdots & \vdots \\ p & p \end{bmatrix}.$$

Rotate this factorial design clockwise in the plane by an angle w:

$$\begin{bmatrix} 1\cos(w) + 1\sin(w) & -1\sin(w) + 1\cos(w) \\ \vdots & \vdots \\ 1\cos(w) + p\sin(w) & -1\sin(w) + p\cos(w) \\ 2\cos(w) + 1\sin(w) & -2\sin(w) + 1\cos(w) \\ \vdots & \vdots \\ 2\cos(w) + p\sin(w) & -2\sin(w) + p\cos(w) \\ \vdots & \vdots \\ p\cos(w) + 1\sin(w) & -p\sin(w) + 1\cos(w) \\ \vdots & \vdots \\ p\cos(w) + p\sin(w) & -p\sin(w) + p\cos(w) \end{bmatrix}.$$

The design matrix (adding an intercept) for this rotated factorial design is

$$X = \begin{bmatrix} 1 & 1\cos(w) + 1\sin(w) & -1\sin(w) + 1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & 1\cos(w) + p\sin(w) & -1\sin(w) + p\cos(w) \\ 1 & 2\cos(w) + 1\sin(w) & -2\sin(w) + 1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & 2\cos(w) + p\sin(w) & -2\sin(w) + p\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & p\cos(w) + 1\sin(w) & -p\sin(w) + 1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & p\cos(w) + p\sin(w) & -p\sin(w) + p\cos(w) \end{bmatrix} = \begin{bmatrix} X_0 & X_1 & X_2 \end{bmatrix}.$$

Now center the columns X_1 and X_2, to make them uncorrelated with the intercept, by subtracting off the averages.
Average of X_1 column is $\frac{pp(p+1)/2}{p^2}(\cos(w) + \sin(w)) = \frac{p+1}{2}(\cos(w) + \sin(w))$.

Average of X_2 column is $\frac{p+1}{2}(-\sin(w)+\cos(w))$.
Define $a_i = i - \frac{p+1}{2}$. (Note that $\sum_{i=1}^{p} a_i = 0$ and $a_i = -a_{p+1-i}$.)
Then the centered design matrix is

$$X' = \begin{bmatrix} 1 & a_1\cos(w)+a_1\sin(w) & -a_1\sin(w)+a_1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & a_1\cos(w)+a_p\sin(w) & -a_1\sin(w)+a_p\cos(w) \\ 1 & a_2\cos(w)+a_1\sin(w) & -a_2\sin(w)+a_1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & a_2\cos(w)+a_p\sin(w) & -a_2\sin(w)+a_p\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & a_p\cos(w)+a_1\sin(w) & -a_p\sin(w)+a_1\cos(w) \\ \vdots & \vdots & \vdots \\ 1 & a_p\cos(w)+a_p\sin(w) & -a_p\sin(w)+a_p\cos(w) \end{bmatrix} = \begin{bmatrix} X_0' & X_1' & X_2' \end{bmatrix}.$$

The regression effect estimates will be uncorrelated if the $X'^T X'$ matrix is diagonal; that is, if $X_1'^T X_2' = 0$. (Note that centering the X_1 and X_2 columns ensures that $X_0'^T X_1' = 0$ and $X_0'^T X_2' = 0$.)

$$\begin{aligned} X_1'^T X_2' &= \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j \cos^2(w) - \sum_{i=1}^{p}\sum_{j=1}^{p} a_i a_j \sin^2(w) \\ &\quad + p\sum_{i=1}^{p} a_i^2 \sin(w)\cos(w) - p\sum_{i=1}^{p} a_i^2 \sin(w)\cos(w) \\ &= (\cos^2(w) - \sin^2(w)) \sum_{i=1}^{p} a_i \sum_{j=1}^{p} a_j \\ &= 0 \end{aligned}$$

so that the rotated full factorial design has uncorrelated regression effect estimates.

Now, let X' be the centered design matrix of an unmodified rotated factorial design with uncorrelated regression effect estimates. (A rotated full factorial design is one such design.) Consider removing four points of this design to obtain a different unmodified rotated factorial design with design matrix X''. The four points removed are

$$\begin{bmatrix} 1 & a_1\cos(w)+a_j\sin(w) & -a_1\sin(w)+a_j\cos(w) \\ 1 & a_j\cos(w)+a_p\sin(w) & -a_j\sin(w)+a_p\cos(w) \\ 1 & a_{p+1-j}\cos(w)+a_1\sin(w) & -a_{p+1-j}\sin(w)+a_1\cos(w) \\ 1 & a_p\cos(w)+a_{p+1-j}\sin(w) & -a_p\sin(w)+a_{p+1-j}\cos(w) \end{bmatrix} = \begin{bmatrix} W_0 & W_1 & W_2 \end{bmatrix}.$$

Note then that

$$\begin{aligned} X_0''^T X_1'' &= X_0'^T X_1' - W_0^T W_1 \\ &= 0 - (a_1 + a_j + a_{p+1-j} + a_p)(\cos(w) + \sin(w)) \\ &= 0 \end{aligned}$$

and

$$\begin{aligned} X_0''^T X_2'' &= X_0'^T X_2' - W_0^T W_2 \\ &= 0 - (a_1 + a_j + a_{p+1-j} + a_p)(-\sin(w) + \cos(w)) \\ &= 0 \end{aligned}$$

so that X'' is a centered design matrix. Finally,

$$\begin{aligned}
X_1^{''T} X_2^{''} &= X_1^{'T} X_2' - W_1^T W_2 \\
&= 0 - [(a_1 a_j + a_j a_p + a_{p+1-j} a_1 + a_p a_{p+1-j})(\cos^2(w) - \sin^2(w)) \\
&\quad +(a_j^2 + a_p^2 + a_1^2 + a_{p+1-j}^2)\sin(w)\cos(w) \\
&\quad -(a_1^2 + a_j^2 + a_{p+1-j}^2 + a_p^2)\sin(w)\cos(w)] \\
&= (a_1(a_j + a_{p+1-j}) + a_p(a_j + a_{p+1-j}))(\sin^2(w) - \cos^2(w)) \\
&= (a_1 + a_p)(\sin^2(w) - \cos^2(w)) \\
&= 0
\end{aligned}$$

and the unmodified rotated factorial design with design matrix X'' has uncorrelated regression effect estimates. By induction, any unmodified rotated factorial design has uncorrelated regression effect estimates.

■

B A Sample Construction

To create 12 point rotated factorial designs:

Start with a 4^2 standard factorial design.

$$\begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 1 \\ 2 & 2 \\ 2 & 3 \\ 2 & 4 \\ 3 & 1 \\ 3 & 2 \\ 3 & 3 \\ 3 & 4 \\ 4 & 1 \\ 4 & 2 \\ 4 & 3 \\ 4 & 4 \end{bmatrix}$$

Rotate by $\tan^{-1}(1/4)$. This yields a 16 point rotated factorial design.

$$\begin{bmatrix}
1\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\
1\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\
1\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\
1\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4)) \\
2\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\
2\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\
2\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\
2\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4)) \\
3\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\
3\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\
3\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\
3\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4)) \\
4\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\
4\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\
4\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\
4\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4))
\end{bmatrix} = \begin{bmatrix}
1.21 & 0.73 \\ 1.46 & 1.70 \\ 1.70 & 2.67 \\ 1.94 & 3.64 \\ 2.18 & 0.49 \\ 2.43 & 1.46 \\ 2.67 & 2.43 \\ 2.91 & 3.40 \\ 3.15 & 0.24 \\ 3.40 & 1.21 \\ 3.64 & 2.18 \\ 3.88 & 3.15 \\ 4.12 & 0.00 \\ 4.37 & 0.97 \\ 4.61 & 1.94 \\ 4.85 & 2.91
\end{bmatrix}$$

Remove the 4 most extreme design points. This yields a 12 point Type U rotated factorial design.

$$\begin{bmatrix} 1.46 & 1.70 \\ 1.70 & 2.67 \\ 2.18 & 0.49 \\ 2.43 & 1.46 \\ 2.67 & 2.43 \\ 2.91 & 3.40 \\ 3.15 & 0.24 \\ 3.40 & 1.21 \\ 3.64 & 2.18 \\ 3.88 & 3.15 \\ 4.37 & 0.97 \\ 4.61 & 1.94 \end{bmatrix} = \begin{bmatrix} 1\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\ 1\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -1\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\ 2\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\ 2\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\ 2\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\ 2\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -2\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4)) \\ 3\cos(\tan^{-1}(1/4)) + 1\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 1\cos(\tan^{-1}(1/4)) \\ 3\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\ 3\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \\ 3\cos(\tan^{-1}(1/4)) + 4\sin(\tan^{-1}(1/4)) & -3\sin(\tan^{-1}(1/4)) + 4\cos(\tan^{-1}(1/4)) \\ 4\cos(\tan^{-1}(1/4)) + 2\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 2\cos(\tan^{-1}(1/4)) \\ 4\cos(\tan^{-1}(1/4)) + 3\sin(\tan^{-1}(1/4)) & -4\sin(\tan^{-1}(1/4)) + 3\cos(\tan^{-1}(1/4)) \end{bmatrix}$$

To get a 12 point Type E rotated factorial design, adjust the rotation angle to $\tan^{-1}(1/3)$. Figuring out the correct rotation angle is easy. If the original design has p^2 points, then the angle is unadjusted if 0 points are removed and is adjusted to $\tan^{-1}(1/(p-1))$ if $\{2, 4, \ldots, 2p-2\}$ points are removed or to $\tan^{-1}(1/(p-2))$ if $\{2p, 2p+2, \ldots, 4p-8\}$ points are removed.

$$\begin{bmatrix} 1\cos(\tan^{-1}(1/3)) + 2\sin(\tan^{-1}(1/3)) & -1\sin(\tan^{-1}(1/3)) + 2\cos(\tan^{-1}(1/3)) \\ 1\cos(\tan^{-1}(1/3)) + 3\sin(\tan^{-1}(1/3)) & -1\sin(\tan^{-1}(1/3)) + 3\cos(\tan^{-1}(1/3)) \\ 2\cos(\tan^{-1}(1/3)) + 1\sin(\tan^{-1}(1/3)) & -2\sin(\tan^{-1}(1/3)) + 1\cos(\tan^{-1}(1/3)) \\ 2\cos(\tan^{-1}(1/3)) + 2\sin(\tan^{-1}(1/3)) & -2\sin(\tan^{-1}(1/3)) + 2\cos(\tan^{-1}(1/3)) \\ 2\cos(\tan^{-1}(1/3)) + 3\sin(\tan^{-1}(1/3)) & -2\sin(\tan^{-1}(1/3)) + 3\cos(\tan^{-1}(1/3)) \\ 2\cos(\tan^{-1}(1/3)) + 4\sin(\tan^{-1}(1/3)) & -2\sin(\tan^{-1}(1/3)) + 4\cos(\tan^{-1}(1/3)) \\ 3\cos(\tan^{-1}(1/3)) + 1\sin(\tan^{-1}(1/3)) & -3\sin(\tan^{-1}(1/3)) + 1\cos(\tan^{-1}(1/3)) \\ 3\cos(\tan^{-1}(1/3)) + 2\sin(\tan^{-1}(1/3)) & -3\sin(\tan^{-1}(1/3)) + 2\cos(\tan^{-1}(1/3)) \\ 3\cos(\tan^{-1}(1/3)) + 3\sin(\tan^{-1}(1/3)) & -3\sin(\tan^{-1}(1/3)) + 3\cos(\tan^{-1}(1/3)) \\ 3\cos(\tan^{-1}(1/3)) + 4\sin(\tan^{-1}(1/3)) & -3\sin(\tan^{-1}(1/3)) + 4\cos(\tan^{-1}(1/3)) \\ 4\cos(\tan^{-1}(1/3)) + 2\sin(\tan^{-1}(1/3)) & -4\sin(\tan^{-1}(1/3)) + 2\cos(\tan^{-1}(1/3)) \\ 4\cos(\tan^{-1}(1/3)) + 3\sin(\tan^{-1}(1/3)) & -4\sin(\tan^{-1}(1/3)) + 3\cos(\tan^{-1}(1/3)) \end{bmatrix} = \begin{bmatrix} 1.58 & 1.58 \\ 1.90 & 2.53 \\ 2.21 & 0.32 \\ 2.53 & 1.26 \\ 2.85 & 2.21 \\ 3.16 & 3.16 \\ 3.16 & 0.00 \\ 3.48 & 0.95 \\ 3.79 & 1.90 \\ 4.11 & 2.85 \\ 4.43 & 0.63 \\ 4.74 & 1.58 \end{bmatrix}$$

Once constructed, these designs can be rescaled as desired.

C Generating Rotated Factorial Designs in S-Plus

The following function, rfd.2d, can be used to generate the rotated factorial designs that we have presented in this paper. The function rfd.2d has 1 required argument – n – and 4 optional arguments – type, remove, x1lim, and x2lim. The first (required) argument is the number of runs in the desired design. The second (optional) argument is the type of the design, either “U” or “E” (the default). The third (optional) argument is the points to remove in a modified rotated factorial design–one of “ss” (the default), “sl”, “ls”, or “ll”–where the first s (l) instructs to remove the design point with the smallest (largest) x_1-coordinate and the second s (l) instructs to remove the design point with the smallest (largest) x_2-coordinate. The fourth and fifth (optional) arguments are the limits of the X_1 and X_2 factors, respectively. The default for these last two arguments is from 1 to a suitably chosen value so that all design points have integer coordinates. Some examples of calling rfd.2d are as follows:

To generate a 17 point Type E design

```
rfd.2d(17)
```

```
      [,1] [,2]
 [1,]    1   11
 [2,]    2   15
 [3,]    3    2
 [4,]    4    6
 [5,]    5   10
 [6,]    6   14
 [7,]    7    1
 [8,]    8    5
 [9,]    9    9
[10,]   10   13
[11,]   11   17
[12,]   12    4
[13,]   13    8
[14,]   14   12
[15,]   15   16
[16,]   16    3
[17,]   17    7
```

To generate a 12 point Type U design

```
rfd.2d(12,"U")
```

```
      [,1] [,2]
 [1,]    1    7
 [2,]    2   11
 [3,]    4    2
 [4,]    5    6
 [5,]    6   10
 [6,]    7   14
 [7,]    8    1
 [8,]    9    5
 [9,]   10    9
[10,]   11   13
[11,]   13    4
[12,]   14    8
```

To generate a 12 point Type E design where X_1 ranges from 10 to 20 and X_2 ranges from 50 to 100

```
rfd.2d(12,x1lim=c(10,20),x2lim=c(50,100))
```

```
      [,1] [,2]
 [1,]   10   75
 [2,]   11   90
 [3,]   12   55
 [4,]   13   70
 [5,]   14   85
 [6,]   15  100
 [7,]   15   50
 [8,]   16   65
 [9,]   17   80
```

```
[10,]   18   95
[11,]   19   60
[12,]   20   75
```

To generate a 10 point Type U modified design with largest points removed

```
rfd.2d(10,"U",remove="ll")

      [,1] [,2]
 [1,]    1    7
 [2,]    2   11
 [3,]    4    2
 [4,]    5    6
 [5,]    6   10
 [6,]    8    1
 [7,]    9    5
 [8,]   10    9
 [9,]   11   13
[10,]   13    4
```

Here is the S-Plus function:

```
rfd.2d <- function(n,type="E",remove="ss",x1lim=c(1,npp),x2lim=c(1,npp))
{
# S function to create rotated factorial designs for computer experiments
# Written by Scott D. Beattie, Penn State University, February 1997
# This function comes with absolutely no warranty.  You alone assume all risk in choosing
# to use this function. In no event shall the author or The Pennsylvania State University
# be held liable for losses to you arising from the use or misuse of this function.
# This function is intended to be distributed freely and may be distributed freely only
# if this header--including author, warranty, and liability information--is included and
# remains unchanged.

ir <- function(num)
{as.integer(round(num))}

rotate.design.2d <- function(p,k=0,angle=atan(1/p),remove="ss")
{
square.points <- function(p)
{apply(matrix(2:1,ncol=1),1,slice.index,x=array(0,dim=rep(p,2)))}

rotate.2d <- function(x,angle,units="radians")
{
if(units=="degrees")
  {angle <- angle/180*pi}
x %*% matrix(c(cos(angle),sin(angle),-sin(angle),cos(angle)),ncol=2)
}

#begin function rotate.design.2d
x <- rotate.2d(square.points(p),atan(1/p))
x <- x[order(x[,2])[(1+trunc(k/4)):(dim(x)[1]-trunc(k/4))],]
x <- x[order(x[,1])[(1+trunc(k/4)):(dim(x)[1]-trunc(k/4))],]
if (k%%4 != 0)
  {
```

```
  x <- x[order(x[,2])[(1+ifelse(remove=="ss"||remove=="ls",1,0)):
        (dim(x)[1]-ifelse(remove=="ss"||remove=="ls",0,1))],]
  x <- x[order(x[,1])[(1+ifelse(remove=="ss"||remove=="sl",1,0)):
        (dim(x)[1]-ifelse(remove=="ss"||remove=="sl",0,1))],]
  }
rotate.2d(rotate.2d(x,-atan(1/p)),angle)
}

norm <- function(x,mult,add)
  {(x-min(x))/(max(x)-min(x))*(mult)+add}

#begin function rfd.2d
if (is.numeric(n)==F)
  {
  warning(paste("\"",n,"\""," is being converted to mode numeric",sep=""))
  if (as.numeric(n)=="NA")
    {stop(paste("\"",n,"\""," cannot be converted to mode numeric",sep=""))}
  n <- as.numeric(n)
  }
if (n%%1!=0)
  {
  warning(paste(n,"is not an integer; converting to nearest integer"))
  n <- ir(n)
  }
if (sqrt(n-2)==round(sqrt(n-2)) || n<=1)
  {stop(paste("No design exists with",n,"runs."))}
if (all(type!=c("U","E")))
  {stop("type argument must equal either \"U\" or \"E\"")}
if (all(remove!=c("ss","sl","ls","ll")))
  {stop("remove argument must equal one of \"ss\", \"sl\", \"ls\", or \"ll\"")}
p <- trunc(sqrt(n)) +
     ifelse(trunc(sqrt(n))^2==n,0,ifelse((n-trunc(sqrt(n))^2)%%2==0,2,1))
k <- p^2-n
if (type=="U" || k==0)
  {
  angle <- atan(1/p)
  npp <- p^2 - k/2
  }
else
  {
  angle <- atan(1/ifelse(p>k/2,p-1,p-2))
  npp <- n - (p-1-ifelse((k/2)%%(p-1)==0,p-1,(k/2)%%(p-1)))
  }
ret <- rotate.design.2d(p,k,angle,remove)
cbind(norm(ret[,1],x1lim[2]-x1lim[1],x1lim[1]),norm(ret[,2],x2lim[2]-x2lim[1],x2lim[1]))
}
```