NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXTENDING ORTHOGONAL AND NEARLY ORTHOGONAL LATIN HYPERCUBE DESIGNS FOR COMPUTER SIMULATION AND EXPERIMENTATION

by

Ang Keng-Ern Joshua

December 2006

Thesis Advisor:	Thomas W. Lucas
Second Reader:	Alejandro S. Hernandez

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE		Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE December 2006	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Extending Orthogonal and Nearly Orthogonal Latin Hypercube Designs for Computer Simulation and Experimentation	5. FUNDING NUMBERS
6. AUTHOR(S) Ang Keng-Ern Joshua	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for release; distribution is unlimited	12b. DISTRIBUTION CODE A

13. ABSTRACT (maximum 200 words)

Computational experimentation is an important tool of the military. It provides useful insights at a lower cost of time and money, when compared to physical experiments. Consequently, computational experiments are used to evaluate weapon systems for technology acquisition, examine tactics, and to help select among alternatives for military operations and war plans.

Experiments often consist of a large number of factors. Advancements in computing power and design of experiments (DOE) for simulation allow for the investigation of more of these factors through computational experiments, achieved with less expense in time, effort, and money. Within the framework of DOE, this thesis investigates Orthogonal Latin Hypercube (OLH) and Nearly Orthogonal Latin Hypercube (NOLH) designs. These designs are often used for computational experiments. This research greatly expands upon the size (in terms of runs and, especially, variables) of the available OLH and NOLH designs. Previously, the largest catalogued OLH and NOLH designs were a maximum of 29 variables and 257 runs. OLH and good space-filling NOLH designs for up to 512 variables in 1025 runs are now available. This thesis also develops an algorithm for handling discrete factors with the designs. Finally, the effects of stacking multiple OLH designs into one larger design are quantified. All of the designs developed in this research are available at the Simulation Experiments & Efficient Designs (SEED) Center website (http://harvest.nps.edu).

14. SUBJECT TERMS Design of Experiments, Latin Hypercubes, Orthogonal Latin Hypercubes, Orthogonal, Space-filling, Computer Simulation and Experiments	15. NUMBER OF PAGES 71
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

EXTENDING ORTHOGONAL AND NEARLY ORTHOGONAL LATIN HYPERCUBE DESIGNS FOR COMPUTER SIMULATION AND EXPERIMENTS

Ang Keng-Ern Joshua
Captain, Singapore Armed Forces, Army
B. Eng Honours, Imperial College of Science, Technology and Medicine, U.K., 2001

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
December 2006

Author: Ang Keng-Ern Joshua

Approved by: Thomas W. Lucas
Thesis Advisor

Alejandro S. Hernandez, LTC, USA
Second Reader

James N. Eagle
Chairman, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Computational experimentation is an important tool of the military. It provides useful insights at a lower cost of time and money, when compared to physical experiments. Consequently, computational experiments are used to evaluate weapon systems for technology acquisition, examine tactics, and to help select among alternatives for military operations and war plans.

Experiments often consist of a large number of factors. Advancements in computing power and design of experiments (DOE) for simulation allow for the investigation of more of these factors through computational experiments, achieved with less expense in time, effort, and money. Within the framework of DOE, this thesis investigates Orthogonal Latin Hypercube (OLH) and Nearly Orthogonal Latin Hypercube (NOLH) designs. These designs are often used for computational experiments. This research greatly expands upon the size (in terms of runs and, especially, variables) of the available OLH and NOLH designs. Previously, the largest catalogued OLH and NOLH designs were a maximum of 29 variables and 257 runs. OLH and good space-filling NOLH designs for up to 512 variables in 1025 runs are now available. This thesis also develops an algorithm for handling discrete factors with the designs. Finally, the effects of stacking multiple OLH designs into one larger design are quantified. All of the designs developed in this research are available at the Simulation Experiments & Efficient Designs (SEED) Center website (http://harvest.nps.edu).

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I. INTRODUCTION 1
A. FRAMING THE PROBLEM 1
B. EXPERIMENTS 2
1. Metamodel 3
2. Design of Experiments 4
C. RESEARCH GOALS 5
D. SOFTWARE 6
E. THESIS ORGANIZATION 6

II. DESIGN OF EXPERIMENTS FOR COMPUTER SIMULATION AND EXPERIMENTATION 7
A. LATIN HYPERCUBE DESIGN MATRIX 7
B. LATIN HYPERCUBE SAMPLING 8
C. DESIRABLE CHARACTERISTICS 8
1. Orthogonal Measures 8
a. Absolute Maximum Pairwise Correlation, ρ_{amp} *9*
b. Condition Number, $cond(X^T X)$ *9*
2. Space-Filling Measures 9
a. Modified L_2-Discrepancy, ML_2 10
b. Centered L_2-Discrepancy, CL_2 10
c. Euclidean Maxi-min Distance, Mm 11
D. ORTHOGONAL LATIN HYPERCUBE (OLH) DESIGNS 11
1. Construction of Matrix M for the OLH 11
2. Construction of Matrix S for the OLH 13
3. Construction of Matrix T for the OLH 14
E. NEARLY ORTHOGONAL LATIN HYPERCUBE (NOLH) DESIGNS 15
F. EXTENDING OLH AND NOLH DESIGNS 17

III. DISCRETE/CATEGORICAL VARIABLES AND THEIR EFFECT ON PROPERTIES OF LH DESIGNS 19
A. INTRODUCTION 19
B. METHODOLOGY OF EVALUATION 20
1. Representing Types of Discrete Variables 20
2. Coding Discrete Variables 20
3. Ordering Columns of a LH Design for Additional Discrete Variables 21

IV. RESULTS OF RESEARCH 23
A. SOFTWARE IMPLEMENTATION OF LATIN HYPERCUBE GENERATOR IN JAVA 23
1. Summary of Functions within JAVA Application 24
2. Catalogues of OLH and NOLH Designs 25
B. PROPERTIES OF OLH AND NOLH DESIGNS 27

1. Experimenting and Extending the OLH and NOLH Designs 27
2. Exploring the Construction of Latin Hypercube Designs .. 30
3. Exploring the OLH and NOLH Designs for Discrete Variables 33
4. Exploring the Effects of Stacking on OLH and NOLH Designs 36

V. CONCLUSIONS AND FUTURE RESEARCH 39
A. CONCLUSION OF RESULTS 39
B. FUTURE RESEARCH 39

APPENDIX A. EXTENDING THE ORTHOGONAL LATIN HYPERCUBE FOR M = 4 41

APPENDIX B. CATALOGUE OF ORTHOGONAL LATIN HYPERCUBE DESIGNS 43

APPENDIX C. PROPERTIES OF LATIN HYPERCUBES WITH DIFFERENT CONSTRUCTION 47

LIST OF REFERENCES 49

INITIAL DISTRIBUTION LIST 51

LIST OF FIGURES

Figure 1. Recommended designs from DOE Toolkit (Kleijnen, Sanchez, Lucas, & Cioppa, 2005). 4

Figure 2. Design matrices: (a) A 3-factor, 2-level Full-Factorial Design; (b) A 2-factor Latin Hypercube Design. 7

Figure 3. Illustration of the random and fixed permutation of the **e** vector for construction of the NOLH designs (Best viewed in color). 24

Figure 4. Pairwise plot of the O_8^{17} design. 29

Figure 5. Comparison of speed and ease of construction vs. properties of LH designs (Best viewed in color). 32

Figure 6. Comparison of properties when one *l*-level discrete variable is added to the O_7^{17} design (Best viewed in color). 35

Figure 7. Stacking of O_2^5 to form a $2{:}O_2^5$ design. 36

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1. A permutation matrix $\mathbf{A_1}$ under Ye's construction for m = 4. 12
Table 2. A permutation matrix $\mathbf{A_1A_3}$ under Ye's construction for m = 4. 12
Table 3. A matrix **M** under Ye's construction for m = 4. 13
Table 4. A matrix **S** under Ye's construction for m = 4. 14
Table 5. O_6^{17} under Ye's construction (where m = 4, k = 6, n = 17). 14
Table 6. Comparison for k between Ye and Cioppa. 15
Table 7. Number of possible permutations of vector **e**. 15
Table 8. Comparison of LH designs with p-way combinations. 17
Table 9. Properties of O_6^{17}, O_7^{17} and O_8^{17}. 18
Table 10. Coding methods for nominal variables. 20
Table 11. Coding for ordinal variables. 21
Table 12. Ranking and selection of columns for discrete variables. 22
Table 13. List of LH designs generated. 23
Table 14. Design matrix of O_{31}^{65}. 26
Table 15. Construction of OLH designs with 3-way and 4-way combinations. ... 27
Table 16. Properties of OLH designs (from O_7^{17} to O_{57}^{129}). 28
Table 17. Properties of O_7^{17} and O_8^{17} with total enumeration of **e**. 29
Table 18. Properties of LH designs of different construction (for n = 17)..... 31
Table 19. Problems with Odd-Even coding with addition 2-level discrete variables (demonstrated on O_7^{17}). 34
Table 20. Pos-Neg coding with additional 2-level variables for O_7^{17} and O_7^{16}. 34
Table 21. Properties of $s{:}O_K^n$ for different numbers of stacks. 37

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I want to thank the Lord for guiding me throughout my life, with blessings in every endeavor. To God be the Glory.

I want to thank the Singapore Armed Forces that provided this opportunity to learn at the NPS, to live in Monterey and to love every single moment here.

I want to thank Professor Lucas for his guidance throughout this thesis. His enthusiasm and professionalism benefited me from the initial proposal of studying the Nearly Orthogonal Latin Hypercube designs to the final delivery of this thesis report.

I want to thank COL Hernandez. His invaluable advice through numerous rounds of editing helped improve the contents of the report.

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

The Simulation Experiments & Efficient Designs (SEED) Center of the Naval Postgraduate School (NPS) has a mission to "promote the advancement of collaborative development and the use of simulation and efficient designs to provide decision makers with timely insights on complex systems and operations."

One of the most widely used designs for computational experiments at the SEED Center and elsewhere is the Latin Hypercube (LH) design. LH designs are attractive for computational experiments because they allow analysts to efficiently vary many factors simultaneously. Furthermore, LH designs provide a lot of flexibility for subsequent analysis.

For any given combination of input factors and runs, there exist a large number of possible LH designs. From among the possibilities, LH designs that are orthogonal and with good space-filling properties are preferred. Designs with good orthogonal properties are desirable because they provide uncorrelated estimates for the regression coefficients that analysts frequently use to analyze simulation models. Good space-filling properties are useful for data analysis across the possible range of data because the experimental points are scattered throughout the domain in a uniform-like manner, providing information throughout the design space. LH designs which are orthogonal can have poor space-filling properties. Thus, a special class of LH designs, called Nearly Orthogonal Latin Hypercube (NOLH), which are nearly orthogonal but with improved space-filling properties, were developed by Cioppa (2002).

Existing catalogued Orthogonal Latin Hypercube (OLH) and NOLH designs can handle up to 29 factors in 257 runs. In this thesis, the number and size of designs currently used by the SEED Center have been dramatically increased. Specifically, the readily available OLH and NOLH designs have been

increased to 512 input variables and 1025 design points. Table A shows how this increases from the previous works of Ye (1998) and Cioppa (2002).

m	***N***	***k*** **(Ye)**	***k*** **(Cioppa)**	***k*** **(Ang)**
4	17	6	7	8
5	33	8	11	16
6	65	10	16	32
7	129	12	22	64
8	257	14	29	128
9	513	16	37	256
10	1025	18	46	512

Table A. Maximum number of input variables for OLH and NOLH designs from Ye (1998), Cioppa (2002), and Ang (2006).

By comparing the properties of LH designs with different construction, some of the findings include:

- Florian's method, used in the construction of NOLH designs, improves the orthogonal properties of the LH designs.
- Increasing the combinations of permutation matrices, to increase the number of factors with the same number of runs, tends to cause downgraded orthogonal and space-filling properties.

LH designs assume that the input variables are continuous. The use of discrete variables affects the orthogonal and space-filling properties of the designs. This thesis provides a useful guide for choosing columns for incorporating discrete variables into the catalogued OLH and NOLH designs. This guide provides designs with the best orthogonal and space-filling properties. Some of the findings include:

- Positive-Negative coding is best suited for 2-level variables.
- The orthogonal properties are affected for discrete variables with fewer levels, when compared to discrete variable with more levels.

One efficient way to improve the space-filling of an LH design is to stack several designs together. While this has been done in the past, the effects have not been quantified. This thesis quantifies the improvement of the orthogonal and space-filling properties, in relation to the number of stacks, for the various LH designs. The LH generator developed in the thesis allows for the flexible creation of whatever stacked designs a user desires.

Users of the LH designs can benefit from the larger designs developed, and a clearer understanding of the effects of discrete variables and stacking on the orthogonal and space-filling properties of the LH designs. These designs and the JAVA generator needed to construct new OLH and NOLH designs are available at the SEED Center website (http://harvest.nps.edu).

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. FRAMING THE PROBLEM

Computer simulation and experimentation is important to the military. It is used in the evaluation of weapon systems and technology prior to acquisition, providing the ability to analyze boundary and dangerous conditions that are difficult to replicate or are hazardous in physical experiments. Computer simulation and experimentation of military operations and war plans, with military simulation models like JCATS[1] and MANA[2], allow military decision makers to study the effects of tactics, employment options, mix of forces, weapon acquisition, and testing of war plans. Through computer simulation and experimentation, there are also savings in material, time and money.

Military uses of computer simulation and experimentation range from the acquisition of the 5.56 mm round to the testing of war plans, and each simulation or experiment contains numerous factors. These inputs tend to be uncertain (i.e., the performance of humans and systems in battle) and it is unreliable to use assumptions of strong prior knowledge (Cioppa & Lucas, 2006). Unfortunately, the total enumeration of all possible scenarios for experimentation is an exhaustive task even with today's computing power.

In the field of design of experiments (DOE), there are effective and efficient methods to design computational experiments. Specific DOE for computer simulation and experimentation, an extension from the traditional DOE, has been proposed (Kleijnen, Sanchez, Lucas, & Cioppa, 2005).

Nearly Orthogonal Latin Hypercube (NOLH) designs possess improvements in space-filling properties with nearly orthogonal properties for Latin Hypercube (LH) designs (Cioppa, 2006), (Cioppa & Lucas, 2006). NOLH designs have been commonly used for military simulation experiments since their

---

1 Joint Conflict and Tactical Simulator (JCATS) http://www.jfcom.mil/about/fact_jcats.htm (accessed 27 May 2006).

2 Map Aware Non-uniform Automata (MANA), developed for the New Zealand Army and Defense Force.

inception (See SEED Center website at http://harvest.nps.edu for more details). Current designs readily available include NOLH designs that can handle up to 29 factors with 257 runs.

This thesis extends the catalogue of the designs up to 512 factors with 1025 runs. The effects of using discrete variables with LH designs are also investigated, and methods for obtaining LH designs with good orthogonal and space-filling properties are proposed. The effects of stacking the LH designs on the orthogonal and space-filling properties of the designs are also investigated and quantified.

B. EXPERIMENTS

By conducting experiments through computer simulation and experimentation, one can explore a much greater number of factors than is feasible through traditional physical experimentation. Not only does the use of computer experimentation provide more consistent replications, but it is a great deal more cost efficient than traditional methods of physical experimentation.

Computer simulation and experimentation are used to compare the performance of systems. With traditional experiments, there are numerous methods developed to detect input variables that may or may not cause some of the variation in the responses. This includes randomization, blocking, and replication. In contrast, the use of computer experiments removes the requirement for these three methods, as computer experiments provide identical answers for repeated runs with the same set of inputs, reducing the need for randomization, blocking, and replication (Santer, Williams, & Notz , 2003).

An experiment consists of responses due to two main types of variables, controlled variables and environmental variables. Controlled variables are variables that can be set to “control” the product or process. Environmental variables are the “uncontrollable” conditions under which the product or process is executed. An example of a military experiment is the determination of the accuracy of a tank gun: the controlled variables include the weapon system,

ammunition type, target types and the operating crew; the uncontrolled variables include the weather conditions. The response of the experiment is the distance of the impact point from the aim point.

A metamodel is often used to estimate the effects of the input variables on the responses. The utility of metamodels is that they allow the investigation to focus on the important factors that may influence the experiment. They are then used as quick turn-around estimates of the actual problem and to replace running the actual simulation, which generally takes a long time to generate results. A description of metamodels can be found in subsection 1.

The DOE Toolkit (Kleijnen, Sanchez, Lucas, & Cioppa, 2005), which provides designs that are used to investigate computer simulation and experiments, can be found in summary in subsection 2.

1. Metamodel

A metamodel describes the relationship between the output response (often denoted as *y*) and the *k* input variables (often denoted as $x_1, x_2, ..., x_n$). A metamodel is a function *g* that can be estimated given an experimental design and the corresponding responses, as described in Equation 1.1.

$$y_i = g(x_1, x_2, ..., x_k) \tag{1.1}$$

The multiple regression model[3] is often used as the metamodel for exploration, where the inputs for the regression coefficients $\beta_0, \beta_1, ..., \beta_k$ are usually estimated from computer simulation and experiments by using the method of least squares. The error term, ε, is often assumed to be independent and identically distributed ($i.i.d.$) with expectation, $E(\varepsilon) = 0$, and variance, $Var(\varepsilon) = \sigma^2$.

The multiple regression model with *k* predictor variables is

$$g(x_1, .., x_k) = \beta_0 + \sum_{i=1}^{k} \beta_i x_i + \varepsilon \qquad \text{where } \varepsilon \overset{i.i.d}{\sim} Normal(0, \sigma^2) \tag{1.2}$$

[3] Montgomery, et al. (2001), Chapter 3.

Metamodels with more complex structures, such as the cubic polynomial model, $y = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3 + \varepsilon$ can be represented by Equation 1.2. This is achieved by the assignments $x_1 = x$, $x_2 = x^2$, and $x_3 = x^3$. Similarly, models with interactions, i.e., $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon$, and quadratic polynomial with interactions models, i.e., $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{12} x_1 x_2 + \varepsilon$, can also be represented by Equation 1.2.

2. Design of Experiments

The field of DOE has a rich history in a range of fields, from farming to medicine, and provides for the efficient conduct of experiments. Computer simulation and experimentation provide a vast area in which to expand DOE. Some of the possible designs are proposed by the DOE Toolkit (Kleijnen, Sanchez, Lucas, & Cioppa, 2005) shown in Figure 1:

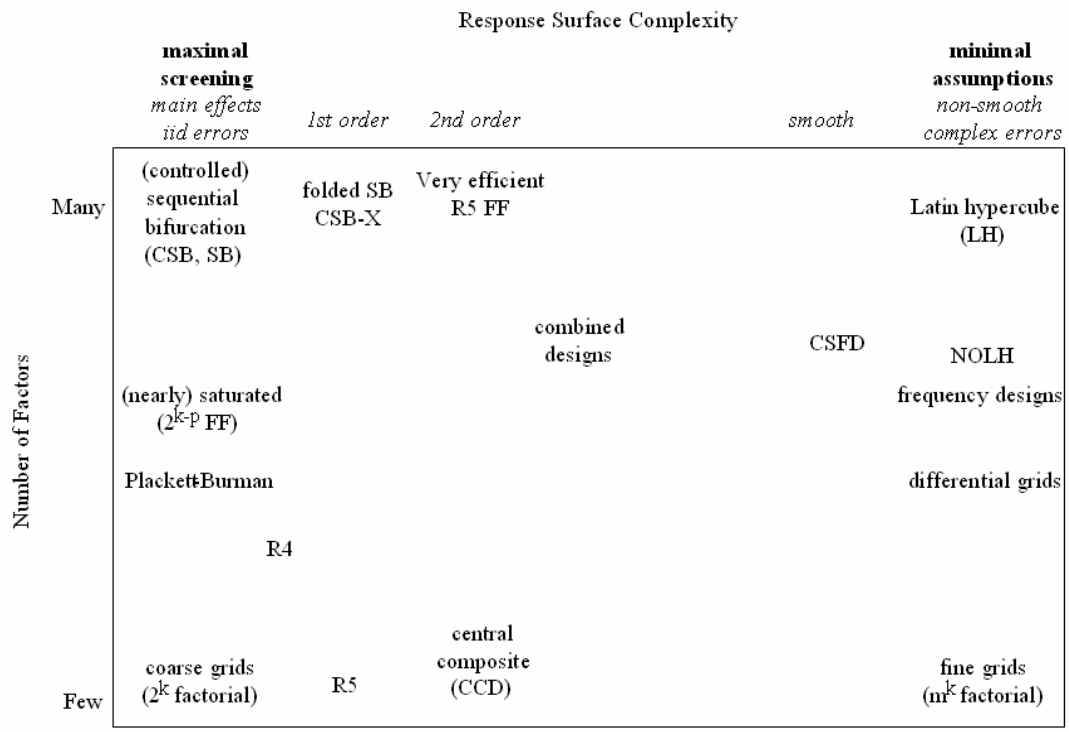

Figure 1. Recommended designs from DOE Toolkit (Kleijnen, Sanchez, Lucas, & Cioppa, 2005).

The number of factors, along the vertical axis, reflects the number of possible "controlled" factors in the simulation experiment. Typical simulation experiments consist of large numbers of potential input factors. The toolkit describes different designs for different applications.

The response-surface complexity, along the horizontal axis, defines the continuum from simple to complex response surfaces, which are affected by the assumptions on the complexity of the metamodel. "Maximal screening" is used when the assumed metamodel is simple, i.e., with no interactions or polynomial terms, and *i.i.d*. errors. "Minimal assumptions" allow for more complex metamodels. The horizontal axis lists these assumptions as guidance for the choice of the designs that can be used.

LH designs are recommended for simulation experiments with large numbers of factors and minimal assumptions. For the assumption of a linear model, Ye developed an algorithm (Ye, 1998) for orthogonal LH designs which were subsequently extended by Cioppa, who exchanged small amounts of non-orthogonality for better space-filling and developed the NOLH designs (Cioppa, 2002). This thesis furthers the current research on these OLH and NOLH designs.

C. RESEARCH GOALS

Construction of the OLH and NOLH designs requires a large amount of computation, computer memory, and time. There are currently a limited number of catalogued designs, and this thesis greatly expands the library of the OLH and NOLH designs to accommodate greater numbers of variables and runs.

Traditional LH designs have been used with the assumption of continuous variables. However, real world variables often do not take on continuous values, rather, they take on discrete values. The orthogonal and space-filling properties of the OLH and NOLH designs are diminished when used with discrete variables. The thesis explores the effects of this, and provides a general rule of thumb when using the LH designs for different numbers and levels of discrete variables.

D. SOFTWARE

JAVA is used as the programming language to generate the LH designs in this thesis. The code was initially developed to be run within the netBeans IDE 5.0[4] running J2SE 5.0. During the course of the development, Eclipse SDK 3.2 for Windows was subsequently used. In addition, an open source COLT package was used to supplement the existing JAVA packages. COLT provides a set of Open Source Libraries for High Performance Scientific and Technical Computing in Java.[5]

With the COLT package, the License Agreement is:

> Copyright (c) 1999 CERN - European Organization for Nuclear Research.
>
> Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

E. THESIS ORGANIZATION

This thesis extends previous work on OLH and NOLH designs. Chapter II focuses on the DOE toolkit, in particular, the orthogonal and space-filling properties of LH designs, and the construction of OLH and NOLH designs. Chapter III investigates the properties of the LH designs with respect to categorical and mixed-level variables. Chapter IV covers the results from the research. Chapter V concludes and recommends future research.

---

[4] netBeans 5.0 and J2SE5.0 Update 6 is available from Sun http://java.sun.com/j2se/1.5.0/download.jsp (last accessed 29 May 2006).

[5] COLT package from http://dsd.lbl.gov/~hoschek/colt/ (last accessed 29 May 2006).

II. DESIGN OF EXPERIMENTS FOR COMPUTER SIMULATION AND EXPERIMENTATION

A. LATIN HYPERCUBE DESIGN MATRIX

A design matrix, **X**, specifies how an experiment is to be conducted, as illustrated in Figure 2. The rows of the matrix, *n*, represent the different design points, or runs, for the experiment. Each row represents a single run of the experiment. The columns, *k*, represent the different factors or controlled variables for the experiment. The $n \times k$ design matrix is used as inputs for the experiment, and the responses of the experiment are recorded as the experiment occurs.

The values in the matrix, represented by (*row*, *column*), state the value for a specific variable during a specific run of the experiment. For example, the representation (*1*, *x1*) reflects that the value of variable *x1* for run 1 is set at one. The notation we use to represent a design matrix is X_k^n.

k variables

n

run	x1	x2	x3
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

(a)

= **X**

run	x1	x2
1	1	2
2	2	1
3	3	4
4	4	3

(b)

Figure 2. Design matrices: (a) A 3-factor, 2-level Full-Factorial Design; (b) A 2-factor Latin Hypercube Design.

A Latin Hypercube (LH) design is represented in Figure 2(b). The structure of LH designs require the columns to be permutations of the integers from $\{1,2,...,n\}$, as shown by X_2^4 in Figure 2(b).

B. LATIN HYPERCUBE SAMPLING

Latin Hypercube Sampling (LHS) is a statistical sampling method to generate inputs for computer experiments (McKay, et al., 1979). The inputs for the computer experiments can be represented as either a Latin Square for two dimensions or an LH design for an arbitrary number of dimensions.

Orthogonal Latin Hypercube (OLH) and Nearly Orthogonal Latin Hypercube (NOLH) designs are special classes of LH designs. OLH designs have strict orthogonal properties, reflected by the condition number and maximum pairwise correlation of the design. Both measures are described in Section C. An OLH design has a condition number of 1 and a maximum pairwise correlation of 0.

The algorithm by Ye (1998) for construction of OLH designs can be found in Section D.

NOLH designs, a term coined by Cioppa (2002), relaxes the requirements on the orthogonal properties. He chooses the most space-filling design amongst a few design matrices that satisfy the near orthogonal thresholds. Typical limits used by Cioppa are a maximum pairwise correlation of no greater than 0.03 and a conditional number no greater than 1.13.

Cioppa's 8-step algorithm to construct NOLH designs can be found in Section E.

C. DESIRABLE CHARACTERISTICS

1. Orthogonal Measures

Designs with good orthogonal properties are desirable because they provide uncorrelated estimates for the regression coefficients that analysts use to analyze simulation models. Two of the orthogonal measures used to examine a

design matrix include the absolute maximum pairwise correlation, ρ_{amp}, and the condition number, $cond(X^T X)$.

a. Absolute Maximum Pairwise Correlation, ρ_{amp}

The absolute maximum pairwise correlation (Hernandez, 2006) is the maximum $|\rho_{ij}|$, over all *i,j,* where $i \neq j$, and ρ_{ij} is the pairwise correlation between columns x_i and x_j. The equation for correlation between any two column vectors (X^i and X^j), where X_l^i and X_l^j denotes the lth entry, while $\bar{X}^i$ and $\bar{X}^j$ are the means of the values in the columns, is given in Equation 2.1.

$$\rho_{ij} = \frac{\sum_{l=1}^{n}\left[\left(X_l^i - \bar{X}^i\right)\left(X_l^j - \bar{X}^j\right)\right]}{\sqrt{\sum_{l=1}^{n}\left(X_l^i - \bar{X}^i\right)^2 \sum_{l=1}^{n}\left(X_l^j - \bar{X}^j\right)^2}} \tag{2.1}$$

ρ_{amp} gives the worst correlation amongst all pairs of columns in the design matrix. A ρ_{amp} of 0 reflects an orthogonal design matrix. As a measure of near orthogonality for a design, a smaller value is preferred.

b. Condition Number, $cond(X^T X)$

For an X_k^n design, $cond(X^T X)=\psi_1/\psi_n$, where ψ_1 and ψ_n are the largest and smallest eigenvalues of $X^T X$, when the columns are centered to sum to zero and scaled to the range [-1,1].

An orthogonal design matrix has a condition number, $cond(X^T X)$, equal to 1. As a measure of near orthogonality for a design, a condition number as close to one as possible is preferred.

2. Space-Filling Measures

Designs with good space-filling properties are useful for data analysis across the possible range of data because the experimental points are scattered throughout the domain in a uniform-like manner. Three measures to assess the

space-filling properties of a design matrix are the modified L_2-discrepancy (ML_2), the centered L_2-discrepancy (CL_2), and the Euclidean maxi-min (Mm) distance.

a. Modified L_2-Discrepancy, ML_2

The L_p-discrepancy is a measure of the non-uniformity of a design (Wei Chen & Sudjianto, 2003). The ML_2 is a variant of the L_p-discrepancy and has been used as a space-filling measure in the construction of NOLH designs (Cioppa, 2002). The measure used by Cioppa is Equation 2.2, and the same measure in the original source (Hickernell, 1998) is Equation 2.3.

$$ML_2(X) = \left(\frac{4}{3}\right)^k - \frac{2^{1-k}}{n}\sum_{d=1}^{n}\prod_{i=1}^{k}(3 - x_{di}^2) + \frac{1}{n^2}\sum_{d=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{k}[2 - \max(x_{di}, x_{ji})] \tag{2.2}$$

$$[ML_2(X)]^2 = \left(\frac{4}{3}\right)^k - \frac{2}{n}\sum_{d=1}^{n}\prod_{i=1}^{k}(3 - x_{di}^2) + \frac{1}{n^2}\sum_{d=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{k}[2 - \max(x_{di}, x_{ji})] \tag{2.3}$$

The two equations differ in two ways: the equation 2.3 reflects that ML_2 is a square root of Right Hand Side (R.H.S.) of the equation, and has $\frac{2}{n}$ in place of $\frac{2^{1-k}}{n}$. Equation 2.2 is used in this thesis with the values normalized to $[0,1]$ in each dimension, to allow for consistency with the efforts on NOLH designs (Cioppa, 2002). As a space-filling measure for a design, a smaller value for ML_2 is preferred.

b. Centered L_2-Discrepancy, CL_2

The CL_2 is another variant for the L_p-discrepancy that measures space-filling. The main difference from ML_2, which is anchored at the origin, is that CL_2 anchors at the center of the hypercube. The equation 2.4 (Fang, et al., 2000) is used with the values normalized to $[0,1]$ in each dimension. As a space-filling measure for a design, a smaller value for CL_2 is preferred.

$$[CL_2(X)]^2 = \left(\frac{13}{2}\right)^k - \frac{2}{n}\sum_{d=1}^{n}\prod_{i=1}^{k}(1+\frac{1}{2}\left|x_{di}-0.5\right|-\frac{1}{2}\left|x_{di}-0.5\right|^2)$$
$$+\frac{1}{n^2}\sum_{d=1}^{n}\sum_{j=1}^{n}\prod_{i=1}^{k}(1+\frac{1}{2}\left|x_{di}-0.5\right|+\frac{1}{2}\left|x_{ji}-0.5\right|-\frac{1}{2}\left|x_{di}-x_{ji}\right|) \quad (2.4)$$

c. *Euclidean Maxi-min Distance, Mm*

Mm is the other space-filling measure used by Cioppa (2002). For values in the designs scaled to the domain $[-1, 1]^k$, a distance list is defined composed of the Euclidean distance between all $\binom{n}{2}$ pairs of design points. *Mm* is the smallest value within the list. The Euclidean distance between any two design points (x_1, x_2) is calculated in equation 2.5 (Morris & Mitchell, 1992). As a space-filling measure, a larger value for *Mm* is preferred.

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^{k}(x_1^{(i)} - x_2^{(i)})^2} \quad (2.5)$$

D. ORTHOGONAL LATIN HYPERCUBE (OLH) DESIGNS

The construction of an $n \times k$ OLH is detailed by Ye (1998). The dimensions of n and k can be determined from m (any positive integer). The number of runs, *n*, is given by $2^m + 1$; the number of variables, *k*, is given by $2m - 2$. OLH designs are denoted as O_k^n.

The creation of an OLH requires the construction of three matrices, **M**, **S** and **T**.

1. Construction of Matrix M for the OLH

The matrix **M**, with dimensions $q \times k$, consists of permutations of the first column, a vector **e** representing the positive levels of a variable. Although vector **e** can be any ordering of the first *q* integers $(1,2,...,q)$, where $q = 2^{m-1}$, an OLH design can be guaranteed when vector **e** is the ascending ordering of integers. For example, when m = 4, q = 2^{4-1} = 8, **e** = $[\,1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8\,]^T$.

Permutation matrices, A_L (created for $L = 1, 2, ..., m-1$) are used to generate the other columns of matrix **M**. Each of the permutation matrices is generated by the following equation, where matrix **I** is the 2 x 2 identity matrix, and matrix $R = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

$$A_L = \underbrace{I \otimes ... \otimes I}_{m-1-L} \otimes \underbrace{R \otimes ... \otimes R}_{L}, \text{ where } \otimes \text{ is the Kronecker product}^{6} \qquad (2.6)$$

For example, with m = 4 and L = 1, the permutation matrix $\mathbf{A_1}$ = I⊗I⊗R (as seen in Table 1) yields the resultant column of $\mathbf{A_1 e}$ = [2 1 4 3 6 5 9 7]^T.

0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0

Table 1. A permutation matrix $\mathbf{A_1}$ under Ye's construction for m = 4.

Additional permutation matrices can be created by multiplying any pair of A_i and A_j, where $i \neq j$. In Ye's algorithm, the columns of M are composed of vector **e**, $\mathbf{A_i e}$ for $i \in \{1, 2, ..., m-1\}$ and $\mathbf{A_i A_{m-1} e}$ for $i \in \{1, 2, ..., m-2\}$. The algorithm uses $m-2$ of the $\binom{m-1}{2}$ 2-way combinations of the set of $\{A_L\}$.

0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0

Table 2. A permutation matrix $\mathbf{A_1 A_3}$ under Ye's construction for m = 4.

[6] Given a *mxn* matrix A and a *pxq* matrix B, the Kronecker product A⊗B would be a (*mxp*)x(*nxp*) matrix C, with each element given by $c_{\alpha\beta} = a_{ij}b_{kl}$ where $\alpha = p(i-1)+k$ and $\beta = q(j-1)+l$. (Source: Weisstein, Eric W. "Matrix Direct Product." From *MathWorld*--A Wolfram Web Resource. http://mathworld.wolfram.com/MatrixDirectProduct.html).

For example, with m = 4, the first addition is the permutation matrix $\mathbf{A_1A_3}$ (as seen in Table 2), which yields the resultant column of $\mathbf{A_1A_3e}$ = [7 8 5 6 3 4 1 2]^T.

An example of matrix **M** for m = 4, n = 17, k = 6, is in Table 3.

e	A_1e	A_2e	A_3e	A_1A_3e	A_2A_3e
1	2	4	8	7	5
2	1	3	7	8	6
3	4	2	6	5	7
4	3	1	5	6	8
5	6	8	4	3	1
6	5	7	3	4	2
7	8	6	2	1	3
8	7	5	1	2	4

Table 3. A matrix **M** under Ye's construction for m = 4.

2. Construction of Matrix S for the OLH

Matrix **S** has entries ± 1, and has the same dimensions as matrix **M**, $q \times k$. The first column of matrix **S** consists of +1's, and the subsequent $m-1$ columns are identical to a two-level full factorial design for $m-1$ variables. The remaining $m-2$ columns are created by pairwise combinations in the two-level full factorial, corresponding to the interactions between $\mathbf{A_L}$ used in the creation of matrix **M**, i.e., $\mathbf{A_1A_3}$. For example, the resultant column, i.e., $\mathbf{S_1S_3}$, is obtained by multiplying, element by element, the columns of $\mathbf{S_1}$ and $\mathbf{S_3}$ together.

Ye proposed the creation of the matrix **S** by defining the vector $\mathbf{a_j}$ as $a_j = B_1 \otimes B_2 \otimes \ldots \otimes B_{m-1}$, where $B_j = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $B_i = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, for $j \in \{1, 2, \ldots, m-1\}$ and $i \neq j$. This vector replaces the columns of S_1, .., S_{m-1}. The subsequent columns are pairwise combinations of the original columns.

For example, with m = 4, the vector $\mathbf{a_1}$ = $\begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ generates the column of $\mathbf{S_1}$ = [-1 +1 -1 +1 -1 +1 -1 +1]^T.

An example of the matrix **S** for m = 4, n = 17, k = 6, is in Table 4.

S_0	S_1	S_2	S_3	S_1S_3	S_2S_3
+ 1	-1	-1	-1	+ 1	+ 1
+ 1	+ 1	-1	-1	-1	+ 1
+ 1	-1	+ 1	-1	+ 1	-1
+ 1	+ 1	+ 1	-1	-1	-1
+ 1	-1	-1	+ 1	-1	-1
+ 1	+ 1	-1	+ 1	+ 1	-1
+ 1	-1	+ 1	+ 1	-1	+ 1
+ 1	+ 1	+ 1	+ 1	+ 1	+ 1

Table 4. A matrix **S** under Ye's construction for m = 4.

3. Construction of Matrix T for the OLH

Matrix **T**, of dimensions $n \times k$, is created by the Hadamard product[7] of matrix **M** and matrix **S**. A row of 0's and a mirror image of matrix **T** are appended to matrix **T** for the creation of an OLH. The mirror for matrix **T** involves changing the sign and flipping the order of the original matrix **T**.

The resultant OLH, O_6^{17}, with **e** = [1 2 3 4 5 6 7 8]^T, is in Table 5.

Run	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6
1	1	- 2	- 4	- 8	7	5
2	2	1	- 3	- 7	- 8	6
3	3	- 4	2	- 6	5	- 7
4	4	3	1	- 5	- 6	- 8
5	5	- 6	- 8	4	- 3	- 1
6	6	5	- 7	3	4	- 2
7	7	-8	6	2	- 1	3
8	8	7	5	1	2	4
9	0	0	0	0	0	0
10	- 8	- 7	- 5	- 1	- 2	- 4
11	- 7	8	- 6	- 2	1	- 3
12	- 6	- 5	7	- 3	- 4	2
13	- 5	6	8	- 4	3	1
14	- 4	- 3	- 1	5	6	8
15	- 3	4	- 2	6	- 5	7
16	- 2	- 1	3	7	8	- 6
17	- 1	2	4	8	- 7	- 5

Table 5. O_6^{17} under Ye's construction (where m = 4, k = 6, n = 17).

Based on the measures defined, we see that the properties for this O_6^{17} are $cond(X^T X)$ = 1.0, ρ_{amp} = 0.0, Mm = 1.4737, ML_2 = 0.0965 and CL_2 = 0.2053.

[7] Suppose ***A*** = (a_{ij}) and ***B*** = (b_{ij}) are to $n \times m$ matrices, the Hadamard product of ***A*** and ***B*** is the entry-wise product of A and B, that is that resultant $n \times m$ matrix has (i,j) entries of $\boldsymbol{a_{ij}b_{ij}}$ (http://planetmath.org/encyclopedia/HadamardProduct.html).

E. NEARLY ORTHOGONAL LATIN HYPERCUBE (NOLH) DESIGNS

Cioppa describes the construction of $n \times k$ OLH and NOLH designs by extending Ye's algorithm (Cioppa, 2002), increasing k from $2m-2$ to $m+\binom{m-1}{2}$, with the number of runs, n, remaining constant. NOLH designs are denoted as N_k^n.

The matrix **M** is extended by using all possible two-way combinations and appending the additional columns, $\mathbf{A_iA_je}$ for $i, j = 1, 2, ..., m-1$ and $i \neq j$. For example, for the case where m = 4, the extension appends the columns, {$\mathbf{A_1A_2e}$, $\mathbf{A_1A_3e}$, $\mathbf{A_2A_3e}$} instead of only {$\mathbf{A_1A_2e}$, $\mathbf{A_2A_3e}$}. This increases the number of variables explored, as depicted in Table 6. Construction of matrix **S** then requires the same number of additional columns, while the construction for matrix **T** remains unchanged.

m	***n***	***k* (Ye)**	***k* (Cioppa)**
4	17	6	7
5	33	8	11
6	65	10	16
7	129	12	22
8	257	14	29

Table 6. Comparison for k between Ye and Cioppa.

By varying the order of **e** in the creation of the matrix **M**, different design matrices, which yield different properties, can be constructed. The original vector **e** = [1 2 3 4 5 6 7 8]^T provides an orthogonal design, but tends to have poor space-filling properties. There are a total of $(2^{m-1})!$ permutations to vector **e**. This creates problems of searching for the best, as seen in the large numbers in Table 7.

m	***Number of Permutations***
4	40320
5	20922789888000
6	263130836933694000000000000000000000

Table 7. Number of possible permutations of vector **e**.

The 8-step construction proposed by Cioppa is detailed (Cioppa & Lucas, 2006) as follows:

Step 1: Determine the number of variables (k > 7) required. If the number of variables is other than 11, 16, 22, or, more generally, ($m+\binom{m-1}{2}$), round the required number of variables up to the nearest one of these numbers.

Step 2: Establish a maximum threshold pairwise correlation value and a maximum threshold condition number. Based on extensive experimentation, we use $\rho_{\max}$ = 0.05, 0.17, 0.16, and $cond(X^T X)$ = 1.15, 2.4, 2.8 for k = 11, 16, and 22, respectively.

Step 3: Using a randomly permuted **e**, construct a design matrix.

Step 4: Calculate the pairwise correlations and the condition number of the candidate matrix.

Step 5: If either value in Step 4 exceeds the thresholds in Step 2, discard the design and return to Step 3 to regenerate with another randomly permuted **e**. Otherwise, keep the design and proceed to Step 6. Repeat Steps 3-5 until a desired number of candidate designs are found. Note: If not enough are found, relax the criteria in Step 2 and begin again. We have found that 15 candidate designs work well for Steps 6-8.

Step 6: Subject each of the candidate designs to repeated applications of Florian's method to decrease the maximum pairwise correlation and condition number. Stop when no further improvement is made.

Step 7: Calculate the Mm distance and *ML2* discrepancy for each of the Step 6 designs. Rank the designs according to some combination of these measures. We chose the design with the minimum rank sum over the two measures.

Step 8: If a number of variables other than 7, 11, 16, 22, or $m+\binom{m-1}{2}$ is required, construct each possible subset having the appropriate number of columns from the Step 7 design and calculate the Mm distance and *ML2* discrepancy. Choose the design with the best combination of these two measures.

With preset thresholds of $cond(X^T X) < 1.13$ and $\rho_{\max} < 0.03$, Cioppa identifies 15 designs that satisfy the definition of near orthogonality. The one with the best space-filling property is chosen (as in Step 7) and catalogued as the best NOLH design, N_k^n, for the particular n and k values.

F. EXTENDING OLH AND NOLH DESIGNS

With Cioppa extending Ye's algorithm to include all 2-way combinations and still maintaining orthogonal properties, the exploration into the inclusion of 3-way and larger combinations for the permutation matrices for construction of the LH designs was performed. Through the analysis, it is confirmed that the orthogonal properties are maintained, thus allowing the ability to investigate a larger number of factors with the same number of design points in an OLH design.

For each design matrix, the inclusion of *p*-way combinations also includes all smaller combinations for the permutation matrices. For example, an inclusion of a 4-way combination also includes both the 2-way and 3-way combinations. The number of factors, *k*, with the inclusion of *p*-way combinations (where the maximum is $p \le m-1$), is given by the following equation:

$$k = 1 + \sum_{j=1}^{p} \binom{m-1}{j} \tag{2.7}$$

The number of factors for each design is improved as shown in Table 8.

n	m	Ye's OLH	Cioppa (2-way)	Ang		
				3-way	4-way	p-way[8]
9	3	4	4	-	-	4
17	4	6	7	8	-	8
33	5	8	11	15	16	16
65	6	10	16	26	31	32
129	7	12	22	42	57	64
257	8	14	29	64	99	128
513	9	16	37	93	163	256
1025	10	18	46	130	256	512

Table 8. Comparison of LH designs with *p*-way combinations.

The construction of the LH designs remains similar to the methods proposed by Ye, and also by Cioppa. The additional permutation matrices include all the combinations of $A_{x_1}A_{x_2}...A_{x_p}$ for $x_1, x_2, .., x_p \in 1, 2, ..., m-1$, and $x_1 \ne x_2 \ne ... \ne x_p$. For example, for *p* = 3, the *2*-way combinations $\mathbf{A_i A_j}$, for $i \ne j$,

[8] The maximum possible *p*-way combination depends on the value of m, where $p = m-1$. For example, if m = 4, the maximum *p*-way combination is a 3-way combination.

and *3*-way combination $\mathbf{A_iA_jA_k}$ are included for the permutation matrices, for all i, j, k $\in$ 1,2,...,m-1 and i $\neq j \neq k$.

For m = 4, the extension appends the column, {$\mathbf{A_1A_2A_3e}$} in addition to the 2-way combinations. Construction of matrix **S** includes the additional number of columns in a similar fashion. The construction of matrix **T** remains unchanged. The details for the construction of the design matrix O_8^{17} can be seen in Appendix A.

With the inclusion of additional combinations for more factors, there are tradeoffs in the orthogonal and space-filling properties for the individual designs. For orthogonal properties, the extended O_8^{17} has the same properties as the methods by Ye and Cioppa. The space-filling properties for the designs with larger numbers of combinations will be worse when compared with the other designs, because we are putting the same number of design points in a larger space. A table of comparison is shown in Table 9.

	$cond(X^TX)$	ρ_{amp}	Mm	ML_2	CL_2
Ye O_6^{17}	1.0	0.0	1.4737	0.0965	0.2053
Cioppa O_7^{17}	1.0	0.0	1.479	0.1732	0.2551
Ang O_8^{17}	1.0	0.0	1.7854	0.3144	0.3142

Table 9. Properties of O_6^{17}, O_7^{17} and O_8^{17}.

For designs with a fixed number of runs, *n*, there are usually tradeoffs for increasing the number of variables, *k*, for the design. The increase in *k*, increases the dimensionality of the design matrix, and improves *Mm*. However, the ML_2 and CL_2 *measures* reflect worse space-filling properties for a larger *k*. For NOLH designs, the limits used are $cond(X^TX) < 1.2$ and $\rho_{amp} < 0.05$.

III. DISCRETE/CATEGORICAL VARIABLES AND THEIR EFFECT ON PROPERTIES OF LH DESIGNS

A. INTRODUCTION

A discrete variable is a variable that assumes values within a discrete set, such as integers. Discrete variables can either be nominal or ordinal variables. Nominal variables are variables where order does not matter, such as gender. Ordinal discrete variables have an ordering or quantitative value associated, e.g., the number of tanks and military ranks.

An example of a nominal discrete variable is gender. Male and Female are often represented in designs in the continuous range by values 1 and 0, or 0 and 1, respectively, as there is no correlation between the representation and the nominal variable. Another example of a nominal discrete variable with more than two levels is the military service, which include Army, Navy, and Air Force. In general, nominal discrete variables with α levels require $\alpha - 1$ indicator variables.

An example of an ordinal discrete variable is the number of soldiers or tanks, which can take only integer values. Another example of an ordinal discrete variable is the military ranks for the Singapore Army, which have an ordinal scale.

Our investigation of the effects of discrete variables on the properties of the LH designs seeks to answer the following questions:

- How does the choice of the column used for the discrete variable affect the properties of the LH design?
- What are the effects on the properties of the LH design for increasing numbers of discrete variables?
- How do different levels for discrete variables affect the properties of the LH design?

- How does different coding of discrete variables affect the properties of the LH design?

B. METHODOLOGY OF EVALUATION

The methodology used to investigate the effects of discrete variables on the properties of LH designs looks at the effects of representing, coding and ordering of discrete variables. OLH and NOLH design matrices constructed earlier are used, and the results of this evaluation are discussed in Chapter IV.

1. Representing Types of Discrete Variables

Both nominal and ordinal discrete variables are investigated. Ordinal discrete variables are more significant, as the presence of ordinal variables is extremely common in real world problems. This thesis looks at the coding methods for both nominal and ordinal discrete variables in the following sections.

2. Coding Discrete Variables

Coding of nominal discrete variables with {high, low} can be done in (at least) two different methods, because the ordering of the values does not matter. The two methods we look at are positive-negative coding and odd-even coding. Positive-negative coding assigns all positive values in our earlier LH designs with the high value and negative values with the low value. Odd-even coding assigns all even numbers with the high value and the odd numbers with the low value.

For example, both coding for the O_7^{17} design are shown in Table 10.

Original	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
Positive-Negative	lo	lo	lo	Lo	lo	lo	lo	lo	**hi**	**hi**	**hi**	**hi**	**hi**	**hi**	**hi**	**hi**	**hi**
Odd-Even	**hi**	lo	**hi**	Lo	**hi**	lo	**hi**	lo	**hi**	lo	**hi**	lo	**hi**	lo	**hi**	lo	**hi**

Table 10. Coding methods for nominal variables.

The mathematical formulation of the positive-negative (PN) and odd-even OD) coding are:

$$coding_{PN} = \begin{cases} \max\ value & if\ x \geq 0 \\ \min\ value & if\ x < 0 \end{cases}$$

$$remainder = MOD(value, 2)$$

$$coding_{OE} = \begin{cases} \max \text{ value} & if\ remainder = 0 \\ \min \text{ value} & if\ remainder \neq 0 \end{cases}$$

Coding discrete variables with more than 2 levels is done by distributing the levels uniformly across the number of levels of the discrete variables. For example, the coding for the 3-level ordinal discrete variable {0, 1, 2} in the O_6^{17} design is represented in Table 11.

Original	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8
Coding	0	0	0	0	0	0	1	1	1	1	1	2	2	2	2	2	2

Table 11. Coding for ordinal variables.

3. Ordering Columns of a LH Design for Additional Discrete Variables

For a single discrete variable, the best column to represent the discrete variable is obtained after comparing the orthogonal and space-filling properties of the design with the discrete variable coded. All of the columns are used for the single discrete variable to generate additional designs. The properties for each new design are ranked, and the column which gives the best properties should be used to represent the single discrete variable.

For example, the O_7^{17} design is used to represent a single 2-level discrete variable. After generating 7 new designs, the properties of the new designs are collected and ranked, as seen in Table 12. The best of the new designs (with the lowest overall rank) is the choice of the column for the 2-level discrete variable. Column 5 is the best choice for O_7^{17} to represent the 2-level discrete variable.

Col. Used	*Raw results*				Ranking				
	$cond(X^T X)$	ρ_{amp}	Mm	ML_2	$cond(X^T X)$	ρ_{amp}	Mm	ML_2	Overall Rank
1	4.004	0.385	1.431	0.627	4	2	3	1	10
2	4.004	0.385	1.392	0.629	4	2	4	2	12
3	4.004	0.385	1.458	0.640	4	2	2	4	12
4	4.004	0.385	1.275	0.665	4	2	6	7	19
5	2.946	0.193	1.474	0.631	1	1	1	3	5 (Best)
6	3.710	0.385	1.341	0.648	2	2	5	6	15
7	3.928	0.385	1.192	0.645	3	2	7	5	17

Table 12. Ranking and selection of columns for discrete variables.

The selection of columns for new LH designs for multiple discrete variables will choose the best column, and sequentially choose the other columns that give the best orthogonal and space-filling properties. An iterative process determines the columns, and although this process may not provide the best (or global optimal) result, the saving of computational time promotes its use over the total enumeration of all possibilities.

The ordering of columns to use for the designs, for additional 2-level variable columns for the O_7^{17} design, is {5, 2, 6, 3, 7, 4, 1}. An example of interpreting this result for a scenario with two 2-level discrete variables using the O_7^{17} design is that we should use columns 5 and 2, because they provide the best orthogonal and space-filling properties. This is further described with illustration in Chapter IV, Tables 19 and 20.

IV. RESULTS OF RESEARCH

A. SOFTWARE IMPLEMENTATION OF LATIN HYPERCUBE GENERATOR IN JAVA

Within the IDE[9], running primarily on a personal laptop[10] or lab computer[11], OLH and NOLH designs corresponding to Table 13 were successfully generated. Properties of the individual designs are discussed in section B. The largest OLH design, O_{512}^{1025}, using $\mathbf{e} = [\ 1 \quad 2 \quad \ldots \ 512\ ]^T$, took approximately 30 minutes to generate. The time taken to generate different NOLH designs is largely dependent on the number of candidate designs considered.

n	m	Additional *p*-way combinations for the permutation matrix							
		2-way	3-way	4-way	5-way	6-way	7-way	8-way	9-way
17	4	7	8	NA	NA	NA	NA	NA	NA
33	5	11	15	16	NA	NA	NA	NA	NA
65	6	16	26	31	32	NA	NA	NA	NA
129	7	22	42	57	63	64	NA	NA	NA
257	8	29	64	99	120	127	128	NA	NA
513	9	37	93	163	219	247	255	256	NA
1025	10	46	130	256	382	466	502	511	512

Table 13. List of LH designs generated.

Inclusion of *p*-way combinations of the permutation matrices during construction of OLH and NOLH designs can achieve a larger LH design. The catalogue of OLH designs captures all designs, and the catalogue of NOLH designs is still in progress. The catalogues can be found on the SEED Center website (http://harvest.nps.edu); some of the designs are catalogued in Appendix B.

The code for the generation of OLH and NOLH is also available on the SEED Center website.

[9] Eclipse SDK (Version: 3.2.0) Build id: M20060629-1905 (Available via: http://www.eclipse.org/downloads/). Last date accessed, Nov 2006

[10] Home laptop – Fujitsu running Intel® Pentium® M 1.73 GHz, 1.00GB RAM.

[11] Lab computer – Dell running Intel® Pentium® 4 3.40 GHz, 2.00 GB RAM.

Construction of NOLH designs are done in two ways, with varying performances in speed and properties of LH designs, namely random and sequential. The random permutation of the vector **e**, generates NOLH design in slightly longer time, but with good properties. The sequential method searches around a known OLH design and generates the NOLH design much faster, but its properties are dependent on the initial starting point. An illustration of the idea for both methods is shown in Figure 3.

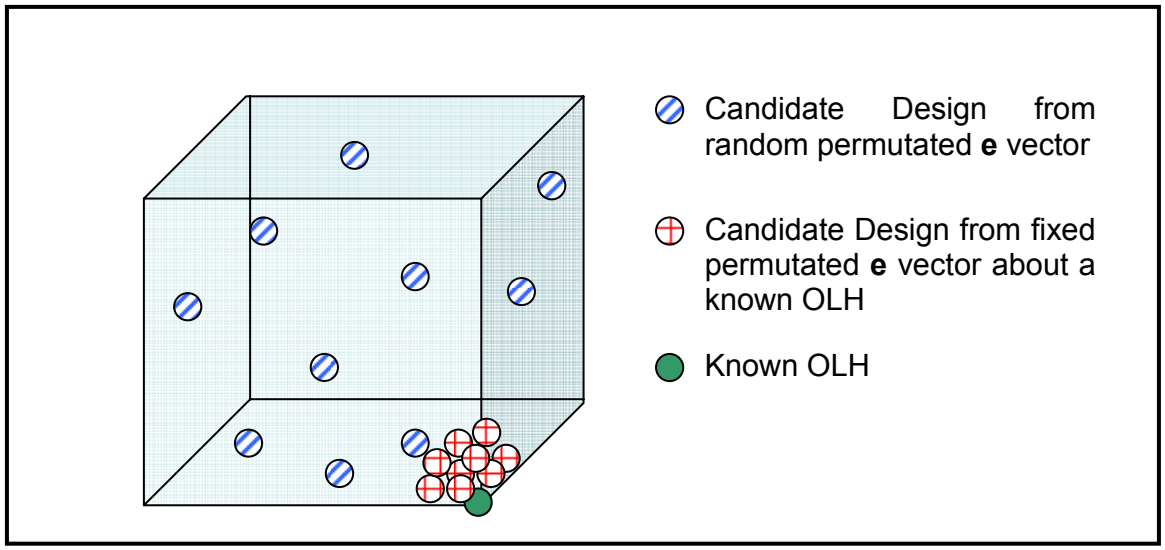

Figure 3. Illustration of the random and fixed permutation of the **e** vector for construction of the NOLH designs (Best viewed in color).

1. Summary of Functions within JAVA Application

The JAVA application is able to do the following:

- Construct OLH to O_2^5 to O_{512}^{1025}.
- Construct NOLH from N_2^5 to N_{93}^{513} (work is still ongoing as 2^{100} "is forever").
- Read in LH design (in .txt format) and output the orthogonal and space-filling properties of the design.

- Read in LH design (in .txt format) and perform a categorical analysis on the design matrix, and output the order of the columns to use which give good properties to the now, mixed design.
- Read in LH design (in .txt format) and stack the design based on user's input, and output the design and its corresponding orthogonal and space-filling properties.

2. Catalogues of OLH and NOLH Designs

A partial catalogue of OLH designs is included in Appendix B. Current designs include the construction for $m \leq 10$ (up to O_{512}^{1025}) and are available on the SEED Center website.

A catalogue of NOLH designs, where the thresholds are $cond(X^T X) < 1.2$ and $\rho_{amp} < 0.05$, is available on the SEED Center website. Construction of NOLH designs will continue to obtain better design matrices, and updated design matrices will also be available.

With the ability to extend the OLH and NOLH designs by the inclusion of additional *p*-way combinations of permutation matrices, the design matrix of O_{31}^{65} is in Table 14.

1	-2	-4	-8	-16	-32	3	7	15	31	5	13	29	9	25	17	-6	-14	-30	-10	-26	-18	-12	-28	-20	-24	11	27	19	23	21
2	1	-3	-7	-15	-31	-4	-8	-16	-32	6	14	30	10	26	18	5	13	29	9	25	17	-11	-27	-19	-23	-12	-28	-20	-24	22
3	-4	2	-6	-14	-30	-1	5	13	29	-7	-15	-31	11	27	19	8	16	32	-12	-28	-20	10	26	18	-22	-9	-25	-17	21	-23
4	3	1	-5	-13	-29	2	-6	-14	-30	-8	-16	-32	12	28	20	-7	-15	-31	11	27	19	9	25	17	-21	10	26	18	-22	-24
5	-6	-8	4	-12	-28	7	-3	11	27	-1	9	25	-13	-29	21	2	-10	-26	14	30	-22	16	32	-24	20	-15	-31	23	-19	-17
6	5	-7	3	-11	-27	-8	4	-12	-28	-2	10	26	-14	-30	22	-1	9	25	-13	-29	21	15	31	-23	19	16	32	-24	20	-18
7	-8	6	2	-10	-26	-5	-1	9	25	3	-11	-27	-15	-31	23	-4	12	28	16	32	-24	-14	-30	22	18	13	29	-21	-17	19
8	7	5	1	-9	-25	6	2	-10	-26	4	-12	-28	-16	-32	24	3	-11	-27	-15	-31	23	-13	-29	21	17	-14	-30	22	18	20
9	-10	-12	-16	8	-24	11	15	-7	23	13	-5	21	-1	17	-25	-14	6	-22	2	-18	26	4	-20	28	32	-3	19	-27	-31	-29
10	9	-11	-15	7	-23	-12	-16	8	-24	14	-6	22	-2	18	-26	13	-5	21	-1	17	-25	3	-19	27	31	4	-20	28	32	-30
11	-12	10	-14	6	-22	-9	13	-5	21	-15	7	-23	-3	19	-27	16	-8	24	4	-20	28	-2	18	-26	30	1	-17	25	-29	31
12	11	9	-13	5	-21	10	-14	6	-22	-16	8	-24	-4	20	-28	-15	7	-23	-3	19	-27	-1	17	-25	29	-2	18	-26	30	32
13	-14	-16	12	4	-20	15	-11	-3	19	-9	-1	17	5	-21	-29	10	2	-18	-6	22	30	-8	24	32	-28	7	-23	-31	27	25
14	13	-15	11	3	-19	-16	12	4	-20	-10	-2	18	6	-22	-30	-9	-1	17	5	-21	-29	-7	23	31	-27	-8	24	32	-28	26
15	-16	14	10	2	-18	-13	-9	-1	17	11	3	-19	7	-23	-31	-12	-4	20	-8	24	32	6	-22	-30	-26	-5	21	29	25	-27
16	15	13	9	1	-17	14	10	2	-18	12	4	-20	8	-24	-32	11	3	-19	7	-23	-31	5	-21	-29	-25	6	-22	-30	-26	-28
17	-18	-20	-24	-32	16	19	23	31	-15	21	29	-13	25	-9	-1	-22	-30	14	-26	10	2	-28	12	4	8	27	-11	-3	-7	-5
18	17	-19	-23	-31	15	-20	-24	-32	16	22	30	-14	26	-10	-2	21	29	-13	25	-9	-1	-27	11	3	7	-28	12	4	8	-6
19	-20	18	-22	-30	14	-17	21	29	-13	-23	-31	15	27	-11	-3	24	32	-16	-28	12	4	26	-10	-2	6	-25	9	1	-5	7
20	19	17	-21	-29	13	18	-22	-30	14	-24	-32	16	28	-12	-4	-23	-31	15	27	-11	-3	25	-9	-1	5	26	-10	-2	6	8
21	-22	-24	20	-28	12	23	-19	27	-11	-17	25	-9	-29	13	-5	18	-26	10	30	-14	6	32	-16	8	-4	-31	15	-7	3	1
22	21	-23	19	-27	11	-24	20	-28	12	-18	26	-10	-30	14	-6	-17	25	-9	-29	13	-5	31	-15	7	-3	32	-16	8	-4	2
23	-24	22	18	-26	10	-21	-17	25	-9	19	-27	11	-31	15	-7	-20	28	-12	32	-16	8	-30	14	-6	-2	29	-13	5	1	-3
24	23	21	17	-25	9	22	18	-26	10	20	-28	12	-32	16	-8	19	-27	11	-31	15	-7	-29	13	-5	-1	-30	14	-6	-2	-4
25	-26	-28	-32	24	8	27	31	-23	-7	29	-21	-5	-17	-1	9	-30	22	6	18	2	-10	20	4	-12	-16	-19	-3	11	15	13
26	25	-27	-31	23	7	-28	-32	24	8	30	-22	-6	-18	-2	10	29	-21	-5	-17	-1	9	19	3	-11	-15	20	4	-12	-16	14
27	-28	26	-30	22	6	-25	29	-21	-5	-31	23	7	-19	-3	11	32	-24	-8	20	4	-12	-18	-2	10	-14	17	1	-9	13	-15
28	27	25	-29	21	5	26	-30	22	6	-32	24	8	-20	-4	12	-31	23	7	-19	-3	11	-17	-1	9	-13	-18	-2	10	-14	-16
29	-30	-32	28	20	4	31	-27	-19	-3	-25	-17	-1	21	5	13	26	18	2	-22	-6	-14	-24	-8	-16	12	23	7	15	-11	-9
30	29	-31	27	19	3	-32	28	20	4	-26	-18	-2	22	6	14	-25	-17	-1	21	5	13	-23	-7	-15	11	-24	-8	-16	12	-10
31	-32	30	26	18	2	-29	-25	-17	-1	27	19	3	23	7	15	-28	-20	-4	-24	-8	-16	22	6	14	10	-21	-5	-13	-9	11
32	31	29	25	17	1	30	26	18	2	28	20	4	24	8	16	27	19	3	23	7	15	21	5	13	9	22	6	14	10	12
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	2	4	8	16	32	-3	-7	-15	-31	-5	-13	-29	-9	-25	-17	6	14	30	10	26	18	12	28	20	24	-11	-27	-19	-23	-21
-2	-1	3	7	15	31	4	8	16	32	-6	-14	-30	-10	-26	-18	-5	-13	-29	-9	-25	-17	11	27	19	23	12	28	20	24	-22
-3	4	-2	6	14	30	1	-5	-13	-29	7	15	31	-11	-27	-19	-8	-16	-32	12	28	20	-10	-26	-18	22	9	25	17	-21	23
-4	-3	-1	5	13	29	-2	6	14	30	8	16	32	-12	-28	-20	7	15	31	-11	-27	-19	-9	-25	-17	21	-10	-26	-18	22	24
-5	6	8	-4	12	28	-7	3	-11	-27	1	-9	-25	13	29	-21	-2	10	26	-14	-30	22	-16	-32	24	-20	15	31	-23	19	17
-6	-5	7	-3	11	27	8	-4	12	28	2	-10	-26	14	30	-22	1	-9	-25	13	29	-21	-15	-31	23	-19	-16	-32	24	-20	18
-7	8	-6	-2	10	26	5	1	-9	-25	-3	11	27	15	31	-23	4	-12	-28	-16	-32	24	14	30	-22	-18	-13	-29	21	17	-19
-8	-7	-5	-1	9	25	-6	-2	10	26	-4	12	28	16	32	-24	-3	11	27	15	31	-23	13	29	-21	-17	14	30	-22	-18	-20
-9	10	12	16	-8	24	-11	-15	7	-23	-13	5	-21	1	-17	25	14	-6	22	-2	18	-26	-4	20	-28	-32	3	-19	27	31	29
-10	-9	11	15	-7	23	12	16	-8	24	-14	6	-22	2	-18	26	-13	5	-21	1	-17	25	-3	19	-27	-31	-4	20	-28	-32	30
-11	12	-10	14	-6	22	9	-13	5	-21	15	-7	23	3	-19	27	-16	8	-24	-4	20	-28	2	-18	26	-30	-1	17	-25	29	-31
-12	-11	-9	13	-5	21	-10	14	-6	22	16	-8	24	4	-20	28	15	-7	23	3	-19	27	1	-17	25	-29	2	-18	26	-30	-32
-13	14	16	-12	-4	20	-15	11	3	-19	9	1	-17	-5	21	29	-10	-2	18	6	-22	-30	8	-24	-32	28	-7	23	31	-27	-25
-14	-13	15	-11	-3	19	16	-12	-4	20	10	2	-18	-6	22	30	9	1	-17	-5	21	29	7	-23	-31	27	8	-24	-32	28	-26
-15	16	-14	-10	-2	18	13	9	1	-17	-11	-3	19	-7	23	31	12	4	-20	8	-24	-32	-6	22	30	26	5	-21	-29	-25	27
-16	-15	-13	-9	-1	17	-14	-10	-2	18	-12	-4	20	-8	24	32	-11	-3	19	-7	23	31	-5	21	29	25	-6	22	30	26	28
-17	18	20	24	32	-16	-19	-23	-31	15	-21	-29	13	-25	9	1	22	30	-14	26	-10	-2	28	-12	-4	-8	-27	11	3	7	5
-18	-17	19	23	31	-15	20	24	32	-16	-22	-30	14	-26	10	2	-21	-29	13	-25	9	1	27	-11	-3	-7	28	-12	-4	-8	6
-19	20	-18	22	30	-14	17	-21	-29	13	23	31	-15	-27	11	3	-24	-32	16	28	-12	-4	-26	10	2	-6	25	-9	-1	5	-7
-20	-19	-17	21	29	-13	-18	22	30	-14	24	32	-16	-28	12	4	23	31	-15	-27	11	3	-25	9	1	-5	-26	10	2	-6	-8
-21	22	24	-20	28	-12	-23	19	-27	11	17	-25	9	29	-13	5	-18	26	-10	-30	14	-6	-32	16	-8	4	31	-15	7	-3	-1
-22	-21	23	-19	27	-11	24	-20	28	-12	18	-26	10	30	-14	6	17	-25	9	29	-13	5	-31	15	-7	3	-32	16	-8	4	-2
-23	24	-22	-18	26	-10	21	17	-25	9	-19	27	-11	31	-15	7	20	-28	12	-32	16	-8	30	-14	6	2	-29	13	-5	-1	3
-24	-23	-21	-17	25	-9	-22	-18	26	-10	-20	28	-12	32	-16	8	-19	27	-11	31	-15	7	29	-13	5	1	30	-14	6	2	4
-25	26	28	32	-24	-8	-27	-31	23	7	-29	21	5	17	1	-9	30	-22	-6	-18	-2	10	-20	-4	12	16	19	3	-11	-15	-13
-26	-25	27	31	-23	-7	28	32	-24	-8	-30	22	6	18	2	-10	-29	21	5	17	1	-9	-19	-3	11	15	-20	-4	12	16	-14
-27	28	-26	30	-22	-6	25	-29	21	5	31	-23	-7	19	3	-11	-32	24	8	-20	-4	12	18	2	-10	14	-17	-1	9	-13	15
-28	-27	-25	29	-21	-5	-26	30	-22	-6	32	-24	-8	20	4	-12	31	-23	-7	19	3	-11	17	1	-9	13	18	2	-10	14	16
-29	30	32	-28	-20	-4	-31	27	19	3	25	17	1	-21	-5	-13	-26	-18	-2	22	6	14	24	8	16	-12	-23	-7	-15	11	9
-30	-29	31	-27	-19	-3	32	-28	-20	-4	26	18	2	-22	-6	-14	25	17	1	-21	-5	-13	23	7	15	-11	24	8	16	-12	10
-31	32	-30	-26	-18	-2	29	25	17	1	-27	-19	-3	-23	-7	-15	28	20	4	24	8	16	-22	-6	-14	-10	21	5	13	9	-11
-32	-31	-29	-25	-17	-1	-30	-26	-18	-2	-28	-20	-4	-24	-8	-16	-27	-19	-3	-23	-7	-15	-21	-5	-13	-9	-22	-6	-14	-10	-12

Table 14. Design matrix of O_{31}^{65}.

B. PROPERTIES OF OLH AND NOLH DESIGNS

1. Experimenting and Extending the OLH and NOLH Designs

Through experimentation, new design matrices to include up to *p*-way combinations of permutation matrices were constructed. These designs have maintained their orthogonal properties of $cond(X^T X)$=1.0 and ρ_{amp} = 0.0, even with the inclusion of 3-way, 4-way, …, $(m-1)$-way combinations when the vector **e** is [1, 2, …, 2^{m-1}]^T. The properties for the LH designs that include 3-way and 4-way combinations are in Table 15, in comparison with the mean of 100 Flexible Random Latin Hypercube (FRLHS) designs.

m	***n***	*3-way*			*4-way*			Mean of 100 FRLHS		
		k	$cond(X^T X)$ ρ_{amp}	***Mm*** ***ML₂***	***k***	$cond(X^T X)$ ρ_{amp}	***Mm*** ***ML₂***	***k***	$cond(X^T X)$ ρ_{amp}	***Mm*** ***ML₂***
4	17	8	**1.0** **0.0**	1.78 0.314	**8**	NA	NA	8	**2.47** **0.24**	1.27 0.35
5	33	15	**1.0** **0.0**	2.20 7.01	**16**	**1.0** **0.0**	2.42 11.52	15	**2.11** **0.17**	1.96 6.23
6	65	26	**1.0** **0.0**	2.70 8.67E2	**31**	**1.0** **0.0**	3.18 1.04E4	26	**1.77** **0.10**	2.77 4.31E2
7	129	42	**1.0** **0.0**	3.27 1.76E6	**57**	**1.0** **0.0**	4.14 5.24E9	42	**1.54** **0.07**	3.27 1.78E5

Table 15. Construction of OLH designs with 3-way and 4-way combinations.

Orthogonal properties of the LH designs are maintained with the inclusion of higher order combinations, but these designs have less ideal space-filling properties. Depending on the type of scenario and the limitations to the number of runs, different design matrices may be used.

After the initial test with additional columns confirmed success in generating OLH designs, additional OLH designs were generated and the properties of the designs are shown in Table 16.

Observations for these designs include:

a) Increasing the number of possible combinations of the permutation matrices increases the number of factors for the design matrices. However, the space-filling nature is less favorable in comparison with designs with fewer combinations of the permutation matrices.

k	n	$cond(X^T X)$	ρ_{amp}	Mm	ML_2	CL_2
7	**17**	1.00	0.00	1.48	0.17	0.26
8	**17**	1.00	0.00	1.79	0.31	0.31
11	**33**	1.00	0.00	1.67	0.95	0.41
15	**33**	1.00	0.00	2.20	7.01	0.73
16	**33**	1.00	0.00	2.42	11.52	0.83
16	**65**	1.00	0.00	1.79	7.98	0.72
22	**129**	1.00	0.00	1.79	96.58	1.39
26	**65**	1.00	0.00	2.70	867.17	2.25
29	**257**	1.00	0.00	1.81	1701.28	2.92
31	**65**	1.00	0.00	3.19	1.E+04	3.816
32	**65**	1.00	0.00	3.34	2.E+04	4.261
37	**513**	1.00	0.00	1.86	5.E+04	6.81
42	**129**	1.00	0.00	3.27	2.E+06	11.34
46	**1025**	1.00	0.00	1.91	2.E+06	18.24
57	**129**	1.00	0.00	4.14	5.E+09	49.27

Table 16. Properties of OLH designs (from O_7^{17} to O_{57}^{129}).

b) The results shown are intuitive. For example, comparing the designs O_{16}^{33} and O_{16}^{65}, shows that a design with more runs, *n*, improves the space-filling property of the design for a fixed number of variables.

c) The orthogonal properties of an OLH design (where $cond(X^T X)$ = 1.0, ρ_{amp} = 0.0) are dependent on the permutation of vector **e**. With a total enumeration of the possible permutations of vector **e** for a 17 run design, there are a total of 640 O_7^{17} and 640 O_8^{17}. These designs result from the same vector **e** that is used in the construction. The total enumeration of larger designs is not possible because the total number of permutations of the vector **e** becomes too big, as shown earlier in Table 7.

Comparing these 640 designs, the properties for the best space-filling designs are in Table 17. The best designs show improvements to the space-filling property, ML_2. A pairwise plot of the best O_8^{17} design is shown in Figure 4.

k	n	$cond(X^T X)$	ρ_{amp}	Mm	ML_2	CL_2	Remarks
7	**17**	1.00	0.00	1.48	0.17	0.26	**e** = [1,2,..,8]
7	**17**	1.00	0.00	1.479	0.1519	0.2426	Best of 640
8	**17**	1.00	0.00	1.79	0.31	0.31	**e** = [1,2,..,8]
8	**17**	1.00	0.00	1.7854	0.2893	0.3003	Best of 640

Table 17. Properties of O_7^{17} and O_8^{17} with total enumeration of **e**.

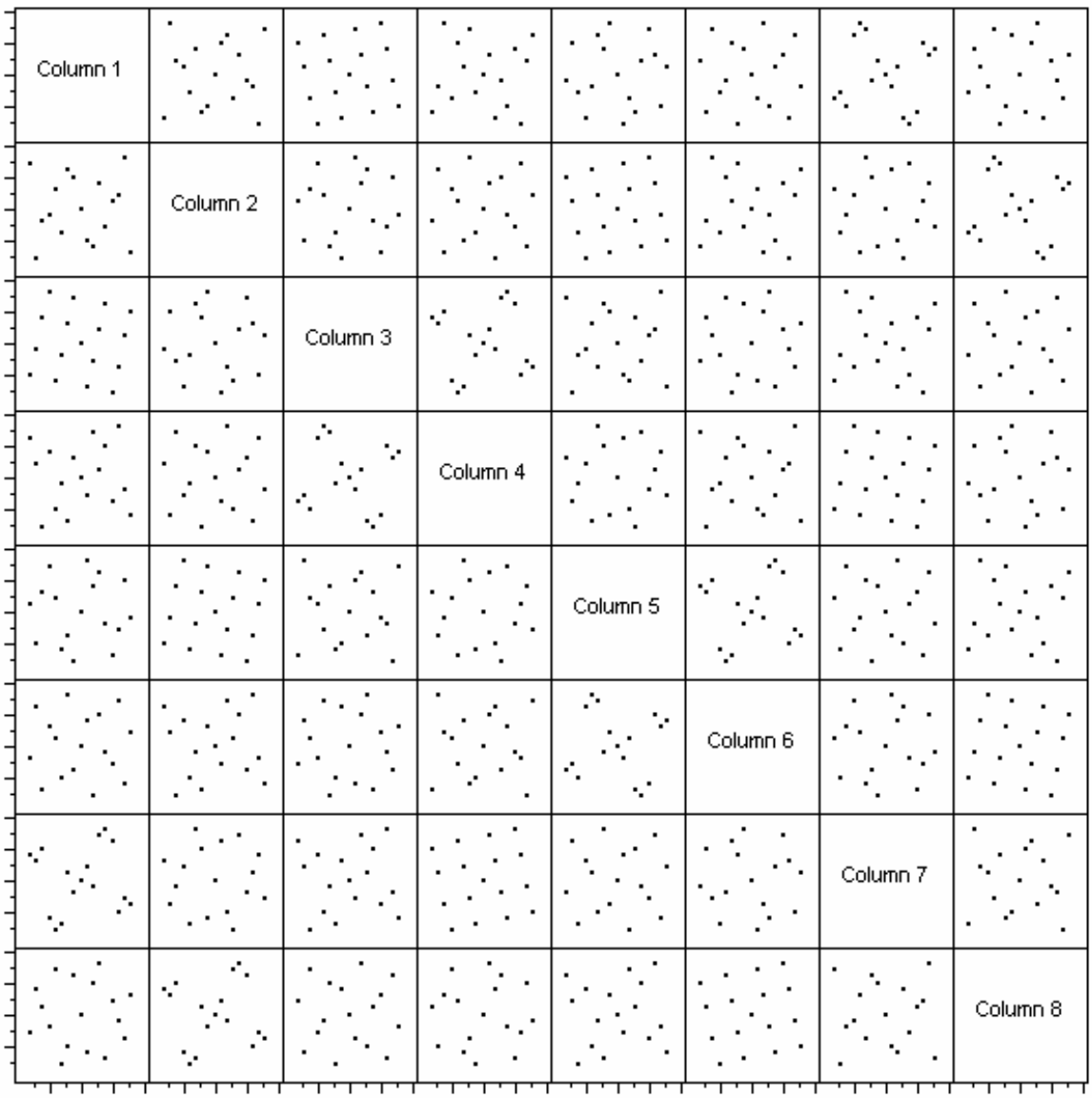

Figure 4. Pairwise plot of the O_8^{17} design.

d) The extension of Ye's algorithm maintains the following two theorems (Ye, 1998):

Theorem 4 (Ye, 1998). *For the LH design constructed, the elementwise square of each column is orthogonal to all the columns in the LH design.*

Theorems 5 (Ye, 1998). *For the LH designs constructed, the elementwise product of every two columns is orthogonal to all columns in the LH design.*

The result of the two theorems provides subsequent estimates of quadratic and bilinear interaction effects that are uncorrelated with the estimates of the linear effects.

2. Exploring the Construction of Latin Hypercube Designs

OLH and NOLH designs can be constructed by different methods, and two methods of construction were compared against catalogued designs. The two methods include random generation and using algorithms proposed by Ye and Cioppa. The driving factors for this investigation are the speed and ease of construction vis-à-vis the orthogonal and space-filling properties of the LH designs.

From the 100 designs generated from Random Latin Hypercube Sampling (RLHS)[12], Florian's method was performed on the RLHS designs, obtaining new designs called Flexible RLHS (FRLHS) designs with improvements to the orthogonal properties. The mean of these 100 designs are used for the comparison.

Designs that were constructed from the algorithm include OLH and NOLH designs and the existing catalogue of designs from the SEED Center website.

[12] RLHS method used MATLAB's X = lhsdesign(n,p) that generates a Latin Hypercube sample X containing n values on each of p variables. The RLHS refers to the average of 100 RLHS designs. FRLHS are Flexible RLHS design that used Florian's method to improve the orthogonal properties.

A subset of the comparison of the properties can be seen in Table 18, with a larger compilation of results seen in Appendix C.

K	***N***	***Type***	$cond(X^T X)$	ρ_{amp}	***Mm***	ML_2
7	**17**	OLH	1.00	0.00	1.48	0.17
		SEED	1.00	0.00	1.48	0.15
		NOLH	1.17	0.05	1.44	0.16
		RLHS	9.53	0.52	0.97	0.57
		FRLHS	2.24	0.23	1.11	0.19
8	**17**	OLH	1.00	0.00	1.79	0.31
		SEED	NA	NA	NA	NA
		NOLH	1.06	0.03	1.79	0.30
		RLHS	14.41	0.57	1.12	0.92
		FRLHS	2.47	0.24	1.27	0.35

Table 18. Properties of LH designs of different construction (for n = 17).

Four properties were considered while comparing the different LH designs: they are the orthogonal properties $cond(X^T X)$ and ρ_{amp}; and the space-filling properties ***Mm*** and ML_2.

The comparison of the properties is based on the results for 50 different designs, using 11 designs (each) from the RLHS, FRLHS, OLH and NOLH designs; and 6 designs from the NOLHdesign_v4.xls (available on the SEED Center website).

In comparing the speed and the ease of construction, the RLHS designs and the NOLH designs from SEED are the easiest and fastest, followed by the FRLHS designs, OLH designs, and NOLH designs.

A parallel plot of the properties for the different LH designs based on different construction methods can be seen in Figure 5.

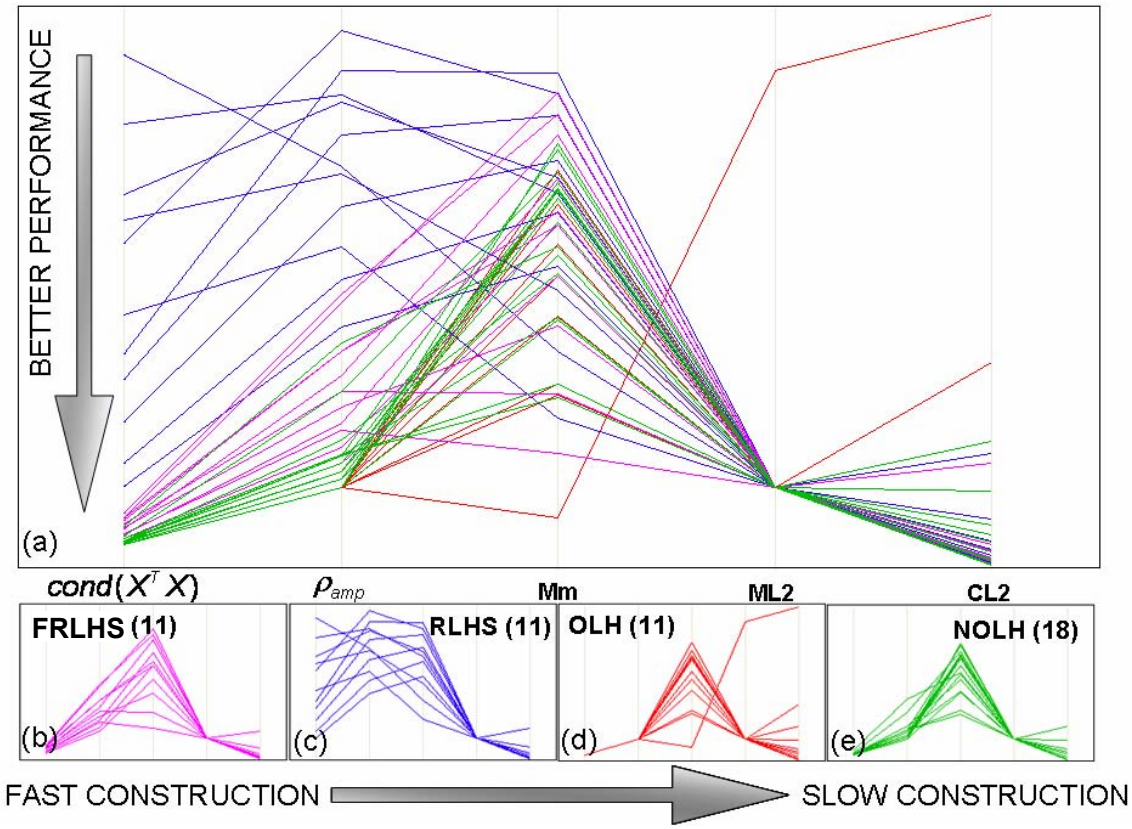

Figure 5. Comparison of speed and ease of construction vs. properties of LH designs (Best viewed in color).

The parallel plot in Figure 5 reflects the improvement of Florian's method on the RLHS to RLHS-F designs, as seen in charts (b) and (c). The odd line of the OLH designs (chart (d)) is due to poor space filling properties of larger designs. OLH and NOLH designs show similar space filling properties, comparable to the average of 100 RLHS and RLHS-F designs. The notable differences between the different construction methods for the LH designs are that:

a. Florian's method improves the orthogonal properties of the design matrices of the RLHS, and the improvement is largely dependent on the *n* and *k* values. The comparison of matrices with similar *n* and *k* reveals that the orthogonal properties are slightly worse than the OLH and NOLH designs. However, RLHS designs have attractive space-filling properties.

b. For smaller design matrices, the space-filling properties do not differ greatly from the OLH with the NOLH or RLHS. However, the ***Mm*** and $\boldsymbol{ML_2}$ measures tend to reflect a downgraded performance with larger design matrices.

c. The comparison between design matrices with the same number of runs, *n*, i.e., including the design matrices with 2-way, 3-way and 4-way combinations of permutation matrices, reveals that the inclusion of higher order combinations of permutation matrices causes downgraded properties for $cond(X^T X)$, ρ_{amp}, and $\boldsymbol{ML_2}$, but an improved ***Mm*** property, as there is a bigger space in which to push the design points apart.

3. Exploring the OLH and NOLH Designs for Discrete Variables

Exploration into the representation of discrete variables, both nominal and ordinal, is important to fitting a scenario into a chosen design matrix. From a user's point of view of a NOLH tool for the design of experiments, having in-depth knowledge on the method of construction for the different designs for different scenarios may not be important. But, given a choice, the user should prefer a better design if available.

The exploration looked into the modifications of an LH design with the inclusion of discrete data, to find the "best" design from the original LH design.

Interesting results were obtained from the study into the properties of OLH and NOLH designs with different numbers and levels of discrete variables.

a. For 2-level discrete variables, Odd-Even coding leads to problems with increased numbers of variables. Although the coding provides for a better orthogonal property with the addition of a single discrete variable with 2-levels, it leads to problems when additional 2-level discrete variables are to be included for the design. The comparison between successive additions of 2-level discrete variables on O_7^{17} with different coding methods is shown on Table 19.

For example, when the O_7^{17} with Odd-Even coding requires three 2-level discrete variables, it has $\rho_{amp} = 1$. The result of $\rho_{amp} = 1$ is also reflected for all other OLH and NOLH designs.

Properties of O7-17 with Pos-Neg Coding						Properties of O7-17 with Odd-Even Coding					
		$cond(X^TX)$	ρ_{amp}	***Mm***	***ML₂***			$cond(X^TX)$	ρ_{amp}	***Mm***	***ML₂***
Original		***1.00***	***0.00***	***1.48***	***0.17***	***Original***		***1.00***	***0.00***	***1.48***	***0.17***
Order of Columns changed	***5***	2.946	0.192	1.474	0.631	***Order of Columns changed***	***3***	2.667	0.000	1.458	0.622
	2	3.817	0.385	1.244	1.062		***1***	16.000	0.889	1.639	1.165
	6	3.042	0.096	1.237	1.483		***6***	**1.6E+17**	**1.000**	1.620	1.868
	3	4.355	0.385	1.125	1.912		***4***	**3.4E+17**	**1.000**	0.612	2.492
	7	3.613	0.192	1.008	2.349		***5***	**1.5E+49**	**1.000**	0.559	3.386
	4	4.743	0.385	1.000	2.840		***2***	**Infinity**	**1.000**	0.250	4.028
	1	1.438	0.056	0.000	3.281		***7***	**Infinity**	**1.000**	0.000	5.070

Table 19. Problems with Odd-Even coding with addition 2-level discrete variables (demonstrated on O_7^{17}).

The O_7^{17} design, with Positive-Negative coding, starts with $cond(X^TX) = 1$ and $\rho_{amp} = 0$ (as seen in Table 19). If one 2-level discrete variable is required, the use of column 5 gives the best properties, as reflected in the chart. If two 2-level discrete variables are required, the use of column 5 and column 2 results in a LH design with better orthogonal and space-filling properties.

Properties of O7-17 with Pos-Neg Coding						Properties of O7-16 with Pos-Negative Coding					
		$cond(X^TX)$	ρ_{amp}	***Mm***	***ML₂***			$cond(X^TX)$	ρ_{amp}	***Mm***	***ML₂***
Original		***1.00***	***0.00***	***1.48***	***0.17***	***Original***		***1.00***	***0.00***	***1.69***	***0.23***
Order of Columns changed	***5***	2.946	0.192	1.474	0.631	***Order of Columns changed***	***5***	2.805	0.198	1.969	0.669
	2	3.817	0.385	1.244	1.062		***2***	3.613	0.396	1.904	1.086
	6	3.042	0.096	1.237	1.483		***6***	2.567	0.099	2.512	1.466
	3	4.355	0.385	1.125	1.912		***1***	3.926	0.396	2.559	1.832
	7	3.613	0.192	1.008	2.349		***4***	2.805	0.198	2.610	2.173
	4	4.743	0.385	1.000	2.840		***3***	3.845	0.396	2.872	2.494
	1	1.438	0.056	0.000	3.281		***7***	1.000	0.000	3.464	2.789

Table 20. Pos-Neg coding with additional 2-level variables for O_7^{17} and O_7^{16}.

Removing the centre row of zeros for O_7^{17} obtains the O_7^{16} design. The effect of additional 2-level discrete variables results in an orthogonal design when all the columns are used to represent 2-level discrete variables (as seen in Table 20). This result holds true for the OLH designs.

b. For ordinal variables, the number of levels, *l*, for the discrete variables affects the performance of the design matrix. Each column will only have values from 1,2, .., *l* if the column is used to represent an *l*-level discrete variable. For example, if representing a 3-level discrete variable, the column used in O_7^{17} will be $[\ 1\ 2\ 3\ 1\ 2\ 3\ 1\ 2\ 3\ 1\ 2\ 3\ 1\ 2\ 3\ 1\ 2\ ]^T$.

*The properties (in part*icular, the orthogonal measures) of O_7^{17} are affected by fewer levels, and the results stabilize as the levels increase. Figure 6 shows the properties for O_7^{17} with different levels of discrete variables, similar for other OLH and NOLH designs.

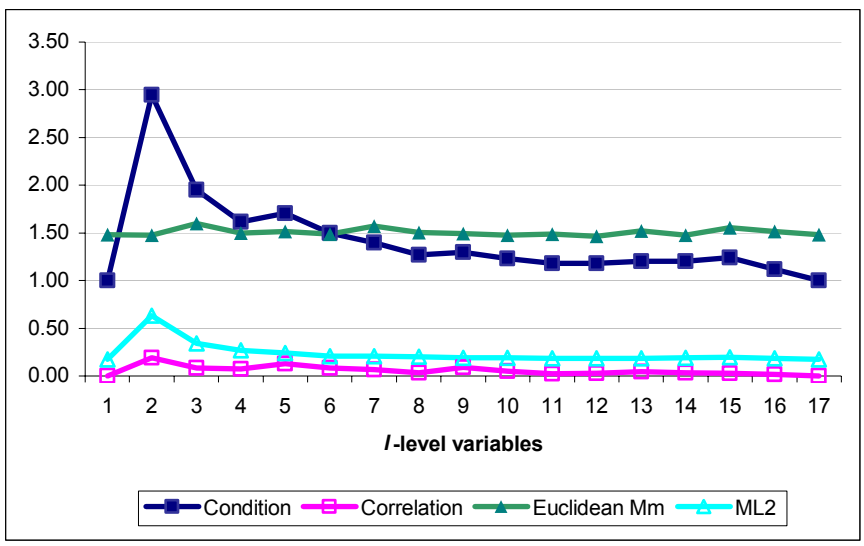

Figure 6. Comparison of properties when one *l*-level discrete variable is added to the O_7^{17} design (Best viewed in color).

c. There is no effect on the orthogonal properties for the addition of discrete variable columns for "complete"[13] designs, which includes O_8^{17}, O_{16}^{33}, O_{32}^{65}, …, O_{512}^{1025}. "Complete" designs do not have variations in the orthogonal properties when the different columns are selected, unlike the smaller designs which result in variation in the orthogonal properties. However, some of the space-filling properties have minor changes. Though with poorer space-filling properties compared to "not complete" designs, this property could be useful to note while considering the design to use, i.e., the use of the OLH and NOLH designs with a larger number of discrete variables.

4. Exploring the Effects of Stacking on OLH and NOLH Designs

Stacking with random permutation of the columns improves the space-filling properties of LH designs. The basic principle behind stacking is that the random permutation of columns maintains the orthogonal properties of an OLH design. Adding additional copies of the designs with random permutation improves the space-filling properties while maintaining orthogonal properties. Furthermore, these are easily constructed and allow replication at each level.

The notation of $s{:}O_k^n$ describes a design with an O_k^n appended with $s-1$ other O_k^n with randomly permuted columns. For example, the illustration in Figure 7 shows the construction of $2{:}O_2^5$.

1	-2
2	1
0	0
-1	2
-2	-1

Original O_2^5 design

1	-2
2	1
-1	2
-2	-1

O_2^5 design (with row of 0s removed)

-2	1
1	2
2	-1
-1	-2

Random permutation of columns of O_2^5 design (with row of 0s removed)

1	-2
2	1
0	0
-1	2
-2	-1
-2	1
1	2
2	-1
-1	-2

$2{:}O_2^5$ design

Figure 7. Stacking of O_2^5 to form a $2{:}O_2^5$ design.

[13] "Complete" designs refer to designs where the construction includes all possible *p*-way combinations of the permutation matrices.

The orthogonal and space-filling properties of the best of five $s{:}O_k^n$ designs are captured in the Table 20.

O_K^n			Number of Stacks (*s*)				
n	**k**	**Properties**	**Original Matrix (1)**	**2**	**5**	**10**	**15**
9	4	**Mm**	**1.369**	0.500	0.354	0.000	0.000
		ML2	**0.054**	0.053	0.054	0.057	0.058
		CL2	**0.184**	0.179	0.173	0.173	0.174
17	8	**Mm**	**1.785**	1.212	0.829	0.530	0.559
		ML2	**0.314**	0.203	0.156	0.129	0.128
		CL2	**0.314**	0.245	0.196	0.171	0.167
33	16	**Mm**	**2.417**	1.620	1.395	1.038	1.104
		ML2	**11.525**	6.324	3.211	2.306	1.828
		CL2	**0.834**	0.605	0.403	0.313	0.262
65	32	**Mm**	**3.342**	2.036	1.952	1.680	1.809
		ML2	**1.72E+04**	1.03E+04	5.22E+03	3.51E+03	2.98E+03
		CL2	**4.261**	3.043	1.943	1.385	1.135
129	64	**Mm**	**4.673**	3.225	3.055	2.660	2.772
		ML2	**3.01E+11**	1.06E+11	4.87E+10	2.96E+10	2.28E+10
		CL2	**98.993**	70.275	44.563	31.554	25.775

Table 21. Properties of $s{:}O_K^n$ for different numbers of stacks.

The points of interest on the effect of stacking to improve the space-filling properties of LH designs include:

a. Random permutation of columns being stacked provides improvements to the space-filling properties. But it does not provide the design with the best properties. Total enumeration is extremely tedious as the total possible permutations is given mathematically as

$$\text{Total permutations} = (k!)\,\hat{}\,s. \tag{4.1}$$

b. For the smaller designs (such as O_2^5 and O_4^9), the *Mm* property becomes 0 due to the randomized permutation repeating the design points due to the stacking of additional matrices. The other space-filling properties of ML_2 and CL_2 show worse performance with an additional number of stacks, possibly due to the same reasons.

c. For the larger designs, although the *Mm* measure becomes worse for all cases, this is due to additional design points being added to the design matrix. The other space-filling properties of ML_2 and CL_2 show improvement in performance with the increased number of stacks.

V. CONCLUSIONS AND FUTURE RESEARCH

A. CONCLUSION OF RESULTS

From the OLH designs proposed by Ye to the NOLH designs by Cioppa, the need for bigger and better designs can be seen. Bigger designs allow for the design of larger computer simulations and experimentation. This, in turn, enables more comprehensive analysis. The inclusion of additional combinations during construction, and the use of JAVA for construction, extends the current catalogues of OLH and NOLH designs to include O_{512}^{1025} and N_{512}^{1025} designs.

The presence of discrete variables is common in most computer simulation and experiments. This thesis provides a useful guide for knowing which columns to change in order to maintain the best orthogonal and space-filling properties; how the effect of different levels affects the properties of the LH design; and, how the use of positive-negative coding is best suited for 2-level nominal variables.

The space-filling properties of LH designs are improved by stacking the designs. The LH generator in JAVA allows for the creation of stacked designs as desired.

B. FUTURE RESEARCH

A subsequent literature review led to a proposed methodology (Steinberg & Lin, 2006) for creating an additional increase in the number of factors for an OLH design. However, the construction method could not be found to access other matrices with similar constructs to compare the space-filling and orthogonal properties. An investigation into this method of construction for the LH designs would be an interesting study, to determine the space-filling properties of the designs in comparison with designs by Ye and Cioppa.

Further analysis of the predictive powers of the extended LH design (which includes both OLH and NOLH designs) is suggested. NOLH designs have orthogonal properties thresholds for selecting candidate designs. It would

be interesting to discover how these limits affect the predictive powers of the designs.

Within the milieu of the JAVA application to generate and access the properties of the LH designs, a smarter and more efficient method to search the space could be investigated to improve the speed of generating the design.

APPENDIX A. EXTENDING THE ORTHOGONAL LATIN HYPERCUBE FOR M = 4

1. Construction of M for the OLH O_8^{17}

The **e** vector used is **e** = [1 2 3 4 5 6 7 8]^T.

For L = 1, **A1e** = [2 1 4 3 6 5 8 7]^T where **A1** = $I \otimes I \otimes R$ yields the following permutation matrix.

0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	0	0	1	0

For L = 2, **A2e** = [4 3 2 1 8 7 6 5]^T where **A2** = $I \otimes R \otimes R$.

For L = 3, **A3e** = [8 7 6 5 4 3 2 1]T where **A3** = $R \otimes R \otimes R$.

Additional permutation matrices are constructed, for example **A1A3** that would obtain **A1A3e** = [7 8 5 6 3 4 1 2]T, yields the following permutation matrix.

0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0

A1A2, gives **A1A2e** = [3 4 1 2 7 8 5 6]^T

A2A3, gives **A2A3e** = [5 6 7 8 1 2 3 4]^T

A1A2A3, gives **A1A2A3e** = [6 5 8 7 2 1 4 3]^T

Thus **M** would be the following matrix.

e	A_1	A_2	A_3	A_1A_2	A_1A_3	A_2A_3	$A_1A_2A_3$
1	2	4	8	3	7	5	6
2	1	3	7	4	8	6	5
3	4	2	6	1	5	7	8
4	3	1	5	2	6	8	7
5	6	8	4	7	3	1	2
6	5	7	3	8	4	2	1
7	8	6	2	5	1	3	4
8	7	5	1	6	2	4	3

2. Construction of S for the OLH O_8^{17}

From the original matrix with S_0, S_1, … , S_3, additional columns are generated. These additional columns are just multiplications of the two columns together, and include S_1S_2, S_1S_3, S_2S_3 and $S_1S_2S_3$.

S_0	S_1	S_2	S_3	S_1S_2	S_1S_3	S_2S_3	$S_1S_2S_3$
+ 1	-1	-1	-1	+ 1	+ 1	+ 1	-1
+ 1	+ 1	-1	-1	-1	-1	+ 1	+ 1
+ 1	-1	+ 1	-1	-1	+ 1	-1	+ 1
+ 1	+ 1	+ 1	-1	+ 1	-1	-1	-1
+ 1	-1	-1	+ 1	+ 1	-1	-1	+ 1
+ 1	+ 1	-1	+ 1	-1	+ 1	-1	-1
+ 1	-1	+ 1	+ 1	-1	-1	+ 1	-1
+ 1	+ 1	+ 1	+ 1	+ 1	+ 1	+ 1	+ 1

3. Construction of T for the OLH O_8^{17}

Hadamard product of **M** and **S**, adding a row of 0s, and the mirror image, results in the design.

Run	Var 1	Var 2	Var 3	Var 4	Var 5	Var 6	Var 7	Var 8
1	1	-2	-4	-8	3	7	5	-6
2	2	1	-3	-7	-4	-8	6	5
3	3	-4	2	-6	-1	5	-7	8
4	4	3	1	-5	2	-6	-8	-7
5	5	-6	-8	4	7	-3	-1	2
6	6	5	-7	3	-8	4	-2	-1
7	7	-8	6	2	-5	-1	3	-4
8	8	7	5	1	6	2	4	3
9	0	0	0	0	0	0	0	0
10	-1	2	4	8	-3	-7	-5	6
11	-2	-1	3	7	4	8	-6	-5
12	-3	4	-2	6	1	-5	7	-8
13	-4	-3	-1	5	-2	6	8	7
14	-5	6	8	-4	-7	3	1	-2
15	-6	-5	7	-3	8	-4	2	1
16	-7	8	-6	-2	5	1	-3	4
17	-8	-7	-5	-1	-6	-2	-4	-3

APPENDIX B. CATALOGUE OF ORTHOGONAL LATIN HYPERCUBE DESIGNS

The designs that are included are as follows.

	Description	Location
	Design Matrix for O_7^{17}	Page 44
	Design Matrix for O_8^{17}	Page 44
	Design Matrix for O_{16}^{33}	Page 45
	Design Matrix for O_{16}^{65}	Page 46
	Other Design Matrix for O_{32}^{65} to O_{512}^{1025}	SEED Center Website http://harvest.nps.edu/

Design Matrix for O_7^{17}

x1	x2	x3	x4	x5	x6	x7
1	-2	-4	-3	8	7	5
2	1	-8	-7	-4	-3	6
8	-4	2	-6	-1	5	-7
4	8	1	-5	2	-6	-3
5	-6	-3	4	7	-8	-1
6	5	-7	8	-3	4	-2
7	-3	6	2	-5	-1	8
3	7	5	1	6	2	4
0	0	0	0	0	0	0
-1	2	4	3	-8	-7	-5
-2	-1	8	7	4	3	-6
-8	4	-2	6	1	-5	7
-4	-8	-1	5	-2	6	3
-5	6	3	-4	-7	8	1
-6	-5	7	-8	3	-4	2
-7	3	-6	-2	5	1	-8
-3	-7	-5	-1	-6	-2	-4

Design Matrix for O_8^{17}

x1	x2	x3	x4	x5	x6	x7	x8
1	-2	-4	-8	3	7	5	-6
2	1	-3	-7	-4	-8	6	5
3	-4	2	-6	-1	5	-7	8
4	3	1	-5	2	-6	-8	-7
5	-6	-8	4	7	-3	-1	2
6	5	-7	3	-8	4	-2	-1
7	-8	6	2	-5	-1	3	-4
8	7	5	1	6	2	4	3
0	0	0	0	0	0	0	0
-1	2	4	8	-3	-7	-5	6
-2	-1	3	7	4	8	-6	-5
-3	4	-2	6	1	-5	7	-8
-4	-3	-1	5	-2	6	8	7
-5	6	8	-4	-7	3	1	-2
-6	-5	7	-3	8	-4	2	1
-7	8	-6	-2	5	1	-3	4
-8	-7	-5	-1	-6	-2	-4	-3

Design Matrix for O_{16}^{33}

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	x13	x14	x15	x16
1	-2	-4	-8	-16	3	7	15	5	13	9	-6	-14	-10	-12	11
2	1	-3	-7	-15	-4	-8	-16	6	14	10	5	13	9	-11	-12
3	-4	2	-6	-14	-1	5	13	-7	-15	11	8	16	-12	10	-9
4	3	1	-5	-13	2	-6	-14	-8	-16	12	-7	-15	11	9	10
5	-6	-8	4	-12	7	-3	11	-1	9	-13	2	-10	14	16	-15
6	5	-7	3	-11	-8	4	-12	-2	10	-14	-1	9	-13	15	16
7	-8	6	2	-10	-5	-1	9	3	-11	-15	-4	12	16	-14	13
8	7	5	1	-9	6	2	-10	4	-12	-16	3	-11	-15	-13	-14
9	-10	-12	-16	8	11	15	-7	13	-5	-1	-14	6	2	4	-3
10	9	-11	-15	7	-12	-16	8	14	-6	-2	13	-5	-1	3	4
11	-12	10	-14	6	-9	13	-5	-15	7	-3	16	-8	4	-2	1
12	11	9	-13	5	10	-14	6	-16	8	-4	-15	7	-3	-1	-2
13	-14	-16	12	4	15	-11	-3	-9	-1	5	10	2	-6	-8	7
14	13	-15	11	3	-16	12	4	-10	-2	6	-9	-1	5	-7	-8
15	-16	14	10	2	-13	-9	-1	11	3	7	-12	-4	-8	6	-5
16	15	13	9	1	14	10	2	12	4	8	11	3	7	5	6
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	2	4	8	16	-3	-7	-15	-5	-13	-9	6	14	10	12	-11
-2	-1	3	7	15	4	8	16	-6	-14	-10	-5	-13	-9	11	12
-3	4	-2	6	14	1	-5	-13	7	15	-11	-8	-16	12	-10	9
-4	-3	-1	5	13	-2	6	14	8	16	-12	7	15	-11	-9	-10
-5	6	8	-4	12	-7	3	-11	1	-9	13	-2	10	-14	-16	15
-6	-5	7	-3	11	8	-4	12	2	-10	14	1	-9	13	-15	-16
-7	8	-6	-2	10	5	1	-9	-3	11	15	4	-12	-16	14	-13
-8	-7	-5	-1	9	-6	-2	10	-4	12	16	-3	11	15	13	14
-9	10	12	16	-8	-11	-15	7	-13	5	1	14	-6	-2	-4	3
-10	-9	11	15	-7	12	16	-8	-14	6	2	-13	5	1	-3	-4
-11	12	-10	14	-6	9	-13	5	15	-7	3	-16	8	-4	2	-1
-12	-11	-9	13	-5	-10	14	-6	16	-8	4	15	-7	3	1	2
-13	14	16	-12	-4	-15	11	3	9	1	-5	-10	-2	6	8	-7
-14	-13	15	-11	-3	16	-12	-4	10	2	-6	9	1	-5	7	8
-15	16	-14	-10	-2	13	9	1	-11	-3	-7	12	4	8	-6	5
-16	-15	-13	-9	-1	-14	-10	-2	-12	-4	-8	-11	-3	-7	-5	-6

Design Matrix for O_{16}^{65}

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	X11	X12	x13	x14	x15	x16
1	-2	-4	-8	-16	-32	3	7	15	31	5	13	29	9	25	17
2	1	-3	-7	-15	-31	-4	-8	-16	-32	6	14	30	10	26	18
3	-4	2	-6	-14	-30	-1	5	13	29	-7	-15	-31	11	27	19
4	3	1	-5	-13	-29	2	-6	-14	-30	-8	-16	-32	12	28	20
5	-6	-8	4	-12	-28	7	-3	11	27	-1	9	25	-13	-29	21
6	5	-7	3	-11	-27	-8	4	-12	-28	-2	10	26	-14	-30	22
7	-8	6	2	-10	-26	-5	-1	9	25	3	-11	-27	-15	-31	23
8	7	5	1	-9	-25	6	2	-10	-26	4	-12	-28	-16	-32	24
9	-10	-12	-16	8	-24	11	15	-7	23	13	-5	21	-1	17	-25
10	9	-11	-15	7	-23	-12	-16	8	-24	14	-6	22	-2	18	-26
11	-12	10	-14	6	-22	-9	13	-5	21	-15	7	-23	-3	19	-27
12	11	9	-13	5	-21	10	-14	6	-22	-16	8	-24	-4	20	-28
13	-14	-16	12	4	-20	15	-11	-3	19	-9	-1	17	5	-21	-29
14	13	-15	11	3	-19	-16	12	4	-20	-10	-2	18	6	-22	-30
15	-16	14	10	2	-18	-13	-9	-1	17	11	3	-19	7	-23	-31
16	15	13	9	1	-17	14	10	2	-18	12	4	-20	8	-24	-32
17	-18	-20	-24	-32	16	19	23	31	-15	21	29	-13	25	-9	-1
18	17	-19	-23	-31	15	-20	-24	-32	16	22	30	-14	26	-10	-2
19	-20	18	-22	-30	14	-17	21	29	-13	-23	-31	15	27	-11	-3
20	19	17	-21	-29	13	18	-22	-30	14	-24	-32	16	28	-12	-4
21	-22	-24	20	-28	12	23	-19	27	-11	-17	25	-9	-29	13	-5
22	21	-23	19	-27	11	-24	20	-28	12	-18	26	-10	-30	14	-6
23	-24	22	18	-26	10	-21	-17	25	-9	19	-27	11	-31	15	-7
24	23	21	17	-25	9	22	18	-26	10	20	-28	12	-32	16	-8
25	-26	-28	-32	24	8	27	31	-23	-7	29	-21	-5	-17	-1	9
26	25	-27	-31	23	7	-28	-32	24	8	30	-22	-6	-18	-2	10
27	-28	26	-30	22	6	-25	29	-21	-5	-31	23	7	-19	-3	11
28	27	25	-29	21	5	26	-30	22	6	-32	24	8	-20	-4	12
29	-30	-32	28	20	4	31	-27	-19	-3	-25	-17	-1	21	5	13
30	29	-31	27	19	3	-32	28	20	4	-26	-18	-2	22	6	14
31	-32	30	26	18	2	-29	-25	-17	-1	27	19	3	23	7	15
32	31	29	25	17	1	30	26	18	2	28	20	4	24	8	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
-1	2	4	8	16	32	-3	-7	-15	-31	-5	-13	-29	-9	-25	-17
-2	-1	3	7	15	31	4	8	16	32	-6	-14	-30	-10	-26	-18
-3	4	-2	6	14	30	1	-5	-13	-29	7	15	31	-11	-27	-19
-4	-3	-1	5	13	29	-2	6	14	30	8	16	32	-12	-28	-20
-5	6	8	-4	12	28	-7	3	-11	-27	1	-9	-25	13	29	-21
-6	-5	7	-3	11	27	8	-4	12	28	2	-10	-26	14	30	-22
-7	8	-6	-2	10	26	5	1	-9	-25	-3	11	27	15	31	-23
-8	-7	-5	-1	9	25	-6	-2	10	26	-4	12	28	16	32	-24
-9	10	12	16	-8	24	-11	-15	7	-23	-13	5	-21	1	-17	25
-10	-9	11	15	-7	23	12	16	-8	24	-14	6	-22	2	-18	26
-11	12	-10	14	-6	22	9	-13	5	-21	15	-7	23	3	-19	27
-12	-11	-9	13	-5	21	-10	14	-6	22	16	-8	24	4	-20	28
-13	14	16	-12	-4	20	-15	11	3	-19	9	1	-17	-5	21	29
-14	-13	15	-11	-3	19	16	-12	-4	20	10	2	-18	-6	22	30
-15	16	-14	-10	-2	18	13	9	1	-17	-11	-3	19	-7	23	31
-16	-15	-13	-9	-1	17	-14	-10	-2	18	-12	-4	20	-8	24	32
-17	18	20	24	32	-16	-19	-23	-31	15	-21	-29	13	-25	9	1
-18	-17	19	23	31	-15	20	24	32	-16	-22	-30	14	-26	10	2
-19	20	-18	22	30	-14	17	-21	-29	13	23	31	-15	-27	11	3
-20	-19	-17	21	29	-13	-18	22	30	-14	24	32	-16	-28	12	4
-21	22	24	-20	28	-12	-23	19	-27	11	17	-25	9	29	-13	5
-22	-21	23	-19	27	-11	24	-20	28	-12	18	-26	10	30	-14	6
-23	24	-22	-18	26	-10	21	17	-25	9	-19	27	-11	31	-15	7
-24	-23	-21	-17	25	-9	-22	-18	26	-10	-20	28	-12	32	-16	8
-25	26	28	32	-24	-8	-27	-31	23	7	-29	21	5	17	1	-9
-26	-25	27	31	-23	-7	28	32	-24	-8	-30	22	6	18	2	-10
-27	28	-26	30	-22	-6	25	-29	21	5	31	-23	-7	19	3	-11
-28	-27	-25	29	-21	-5	-26	30	-22	-6	32	-24	-8	20	4	-12
-29	30	32	-28	-20	-4	-31	27	19	3	25	17	1	-21	-5	-13
-30	-29	31	-27	-19	-3	32	-28	-20	-4	26	18	2	-22	-6	-14
-31	32	-30	-26	-18	-2	29	25	17	1	-27	-19	-3	-23	-7	-15
-32	-31	-29	-25	-17	-1	-30	-26	-18	-2	-28	-20	-4	-24	-8	-16

APPENDIX C. PROPERTIES OF LATIN HYPERCUBES WITH DIFFERENT CONSTRUCTION

k	n	OLH					NOLH (SEED)					Mean of 100 RLHS					Mean of 100 RLHS-F				
		$cond(X^TX)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^TX)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^TX)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^TX)$	ρ_{amp}	*Mm*	*ML2*	*CL2*
7	17	1.00	0.00	1.48	0.17	0.26	1.00	0.00	1.48	0.15	0.24	9.53	0.52	0.97	0.57	0.45	2.24	0.23	1.11	0.19	0.27
8	17	1.00	0.00	1.79	0.31	0.31	-	-	-	-	-	14.41	0.57	1.12	0.92	0.53	2.47	0.24	1.27	0.35	0.33
11	33	1.00	0.00	1.67	0.95	0.41	1.12	0.02	1.76	0.73	0.36	8.44	0.44	1.28	1.94	0.56	1.76	0.14	1.41	0.83	0.38
15	33	1.00	0.00	2.20	7.01	0.73	-	-	-	-	-	16.64	0.48	1.72	11.31	0.96	2.11	0.17	1.96	6.23	0.71
16	33	1.00	0.00	2.42	11.52	0.83	-	-	-	-	-	19.73	0.49	1.83	17.06	1.08	2.24	0.17	2.06	9.94	0.82
16	65	1.00	0.00	1.79	7.98	0.72	1.10	0.02	2.04	4.46	0.54	6.47	0.35	1.60	8.61	0.73	1.45	0.08	1.73	4.80	0.57
22	129	1.00	0.00	1.79	96.58	1.39	1.04	0.01	2.27	37.78	0.86	4.65	0.26	1.97	53.45	1.03	1.27	0.05	2.06	37.26	0.89
26	65	1.00	0.00	2.70	867.17	2.25	-	-	-	-	-	15.50	0.39	2.53	562.75	2.42	1.77	0.10	2.77	431.13	2.08
29	257	1.00	0.00	1.81	2.E+04	2.92	1.02	0.00	2.73	482.91	1.44	3.61	0.20	2.35	488.63	1.61	1.17	0.03	2.42	392.36	1.48
31	65	1.00	0.00	3.19	1.E+04	3.816	-	-	-	-	-	22.84	0.40	2.96	4.E+03	4.33	1.97	0.12	3.26	4.E+03	3.77
32	65	1.00	0.00	3.34	2.E+04	4.261	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
37	513	1.00	0.00	1.86	5.E+04	6.81	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
42	129	1.00	0.00	3.27	2.E+06	11.34	-	-	-	-	-	11.26	0.30	3.44	2.E+05	10.18	1.54	0.07	3.68	2.E+05	9.32
46	1025	1.00	0.00	1.91	2.E+06	18.24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
57	129	1.00	0.00	4.14	5.E+09	49.27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
63	129	1.00	0.00	4.56	1.E+11	89.03	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
64	129	1.00	0.00	4.67	2.E+11	98.99	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
64	257	1.00	0.00	3.88	2.E+11	131.73	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

k	n	OLH					NOLH (SEED)					Mean of 100 RLHS					Mean of 100 RLHS-F				
		$cond(X^T X)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^T X)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^T X)$	ρ_{amp}	*Mm*	*ML2*	*CL2*	$cond(X^T X)$	ρ_{amp}	*Mm*	*ML2*	*CL2*
93	513	1.00	0.00	4.52	1.E+18	4.E+03	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
99	257	1.00	0.00	5.25	3.E+19	5.E+03	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
120	257	1.00	0.00	6.13	2.E+24	3.E+04	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
127	257	1.00	0.00	6.49	9.E+25	6.E+04	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
128	257	1.00	0.00	6.57	1.E+26	7.E+04	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
130	1025	1.00	0.00	5.12	2.E+27	5.E+05	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
163	513	1.00	0.00	6.52	9.E+34	9.E+06	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
219	513	1.00	0.00	8.05	1.E+48	2.E+09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
247	513	1.00	0.00	8.91	3.E+54	2.E+10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
255	513	1.00	0.00	9.21	2.E+56	5.E+10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
256	513	1.00	0.00	9.26	3.E+56	5.E+10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
256	1025	1.00	0.00	7.93	7.E+57	9.E+11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
382	1025	1.00	0.00	10.35	5.E+87	4.E+17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
466	1025	1.00	0.00	11.99	2.E+107	9.E+20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
502	1025	1.00	0.00	12.79	3.E+115	2.E+22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
511	1025	1.00	0.00	13.04	3.E+117	4.E+22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
512	1025	1.00	0.00	13.08	5.E+117	4.E+22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

LIST OF REFERENCES

Cioppa, T.M. (2002). Efficient nearly orthogonal and space-Filling experimental designs for high-dimensional complex models. Doctoral dissertation, U.S. Naval Postgraduate School, Monterey, California.

Cioppa, T.M., & Lucas, T.W. (2006). Efficient nearly orthogonal and space-filling Latin hypercubes. Forthcoming in Technometrics.

Fang, K.T., Ma C.X., & Winker P. (2000). Centered L2-discrepancy of random sampling and Latin hypercube design, and construction of uniform designs. Mathematics of Computation, 71, 275–296.

Hernandez, A.S. (2006). The absolute maximum pairwise correlation of random Latin hypercube designs as a function of sample runs and number of factors. Unpublished manuscript, U.S. Naval Postgraduate School, Monterey, California.

Hickernell, F.J. (1998). A generalized discrepancy and quadrature error bound. Math. Comp., 67, 299-322. Available via http://www.ams.org/mcom/1998-67-221/S0025-5718-98-00894-1/S0025-5718-98-00894-1.pdf, accessed July 26, 2006.

Kleijnen, J.P.C., Sanchez, S.M., Lucas T.W., & Cioppa, T.M. (2005). A user's guide to the brave new world of designing simulation experiments. INFORMS Journal on Computing, 17, 263-289. Available http://www.atypon-link.com/INF/doi/pdf/10.1287/ijoc.1050.0136, accessed July 26, 2006.

Koehler, J.R. & Owen, A.B. (1996), Computer experiments. Available via http://citeseer.ist.psu.edu/cache/papers/cs/6348/http:zSzzSzwww-stat.stanford.eduzSzreportszSzowenzSzmain.pdf/koehler96computer.pdf, accessed July 26, 2006.

McKay, M.D., Beckman, R.J., & Conover, W.J. (1979). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. Technometrics, 21, 239-245.

Montgomery, D. C., Peck, E. A., & Vining, G.G. (2001). Introduction to linear regression analysis. (3rd ed.). New York, NY: John Wiley & sons.

Morris, M.D. & Mitchell, T.J. (1992). Exploratory designs for computational experiments. Available via http://www.ornl.gov/info/reports/1992/3445603672345.pdf, accessed July 26, 2006.

Sanchez, S.M. (2005). Work smarter, not harder: Guidelines for designing simulation experiments. Proceedings of the 2005 Winter Simulation Conference. Available via http://www.informs-cs.org/wsc05papers/009.pdf, accessed July 26, 2006.

Santner, T.J., Williams, B.J., & Notz, W.I. (2003), The design and analysis of computer experiments. Springer-Verlag, New York.

Steinberg, D.M., & Lin D.K.J (2006). A construction method for orthogonal Latin hypercube designs. Biometrika, 93, 279-288.

Tang, B. (1998), Selecting Latin hypercubes using correlation criteria. Statistica Sinica 8, 965-977.

Wei Chen, R.J. & Sudjianto, A. (2003). An efficient algorithm for constructing optimal design of computer experiments. Proceedings of DETC'03. Available via http://ideal.mech.northwestern.edu/pdf/JSPI03.pdf, accessed July 26, 2006.

Ye, K.Q. (1998). Orthogonal column Latin hypercubes and their application in computer experiments. Journal of the American Statistical Association – Theory and Methods, 93, 1430-1439.

Ye, K.Q. (1998b). On the structure of orthogonal Latin hypercubes. Available via http://citeseer.ist.psu.edu/cache/papers/cs/24152/http:zSzzSzwww.ams.sunysb.eduzSz~kyezSzpaperszSzstructure.pdf/ye98structure.pdf, accessed July 26, 2006.

Ye, K., Li, W., & Sudjianto, A. (2006). Algorithmic construction of optimal symmetric Latin hypercube designs. Available via http://citeseer.ist.psu.edu/cache/papers/cs/15527/http:zSzzSzwww.csom.umn.eduzSzWWWPageszSzFacultyzSzWLizSzresearchzSzSLH.pdf/algorithmic-construction-of-optimal.pdf, accessed July 26, 2006.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. LTC Thomas M. Cioppa
   TRADOC Analysis Center – Fort Leavenworth
   Fort Leavenworth, Kansas

4. LTC Jeffery Schamburg
   TRAC Monterey
   Monterey, California

5. LTC Alejandro Hernandez
   Naval Postgraduate School
   Monterey, California

6. Mr. Chwee Seng, Choo
   Defence Science Organization
   Singapore