आई.ऐ.एस.आर.आई/पि.आर.-01 /2019
I.A.S.R.I./P.R.-01/2019
Institute Project Code: AGENIASRISIL201503200069

परियोजना रिपोर्ट
Project Report

लांबिक एवं अंतर प्रबिष्ट लांबिक लैटिन हैपरक्यूब अभिकल्पनाओं के संरचना

On construction of orthogonal and nested orthogonal Latin hypercube designs

सुकान्त दाश	**Sukanta Dash**
बी एन मंडल	**Baidyanath Mandal**
राजेन्द्र प्रसाद	**Rajender Parsad**
सुशील कु सरकार	**Susheel Kumar Sarkar**

परीक्षण अभिकल्पना प्रभाग
Division of Design of Experiments

भा. कृ. अनु. प.–भारतीय कृषि सांख्यिकी अनुसन्धान संस्थान
लाइब्रेरी एवेन्यू, पूसा, नई दिल्ली- 110012
I.C.A.R - Indian Agricultural Statistics Research Institute
Library Avenue, Pusa, New Delhi – 110 012

2019.

<u>आमुख</u>

परीक्षण अभिकल्पना की लैटिन हाइपरक्यूब अभिकल्पनाओं का चयन तब लोकप्रिय है जबकि भौतिक प्रक्रिया के अध्ययन हेतु संगणक अनुकरण का उपयोग लैटिन हाइपरक्यूब के अभिकल्पना बिन्दुओं की तरह होता है जो अभिकल्पना क्षेत्र में समान दूरी पर हाते हैं जब उन्हें सर्वचर सीमा पर प्रक्षेपित किया जाता है। संगणक परीक्षणों में, चरों के स्तरों का परिवर्तन का अर्थ केवल इनपुट के लिए विभिन्न संख्याओं को समायोजित करना है जबकि भौतिक परीक्षणों में, चरों के अणिक स्तर लेन के लिए प्रारूप बनाने के लिए अतिरिक्त कीमत की आवष्यकता व परीक्षण का क्रियान्वयन अधिक लम्बर व समय लगाने वाला हो जाता है। अतः संगणक परीक्षणों एवं परंपरागत भौतिक परीक्षणों में अंतर होन के कारण अभिकल्पना में विभिन्न सोच–विचार एवं संगणक परीक्षणों हेतु विष्लेषण पद्धतियों की आवष्यकमा है । इस प्रकार की परीक्षण परिस्थितियों के संचालन के लिए लाम्बिक लैटिन हाइपरक्यूब अभिकलपना आरंभ की गई। लाम्बिक लैटिन हाइपरक्यूब अभिकल्पना आरंभ की गई। लाम्बिक व्यूह (ओ ए) अभिकल्पनाओं का व्यापक उपयोग परीक्षण नियोजन हेतु किया जाता है और इनकी सफलता समरूपता गुणों के कारण है लेकिन जब किसी परीक्षण में बहुत बड़ी संख्या में कारकों का अध्ययन करना हो और उनमें से बहुत कम वस्तुतः प्रभावी हो, प्रभावी करकों द्वारा पूर्ण विस्तारित उपस्थान पर प्रक्षेपित ओ ए अभिकल्पनाएं परीणाम में केतल प्रभावी हिस्सों पर बिन्दुओं की नुरावृत्ति देगी जो कि भौतिक परीक्षणों जिनमें प्रस्तावित मॉडल का झुकाव प्रसरण से अधिक गंभीर है अनपेक्षित है। इस प्रकरण में लै.हा. अभि. को. वरीयता दी जा सकती है। लेकिन सम द्विचर सीमा पर इस प्रकार के अभिकल्पना बिन्दुओं के प्रक्षेपण की समान रूप से बिखरे होने की गारंटी नहीं दी जा सकती हैं अतः इस प्रकार की परिस्थितियों के संचालन के लिए लाम्बिक व्यूह आधारित लैटिन हाइपरक्यूब प्रस्तावित की गई जिसमें सामान्यतः यादृच्छिक लैटिन हाइपरक्यूब अभिकल्पनाओं की अपेक्षा स्थान भरने के बेहतर गुण हैं। कुछ परीक्षणों में, सत्यनिश्ठा की विभिन्न कोटि पर बड़े और महंगे संगणक कोड और बहुस्तर कीमत और परिषुद्धता वाले संगणक परीक्षण संचालित किए जा सकते हैं। इस प्रकार के परीक्षणों की अभिकल्पनाओं हेतु नीड़ित अभिकल्पनाएं उपयोगी हैं। इस विधि में एल एच डी के उपयोग की मुख्य कती यह है कि इसमें उच्च– परिषुद्धता व निम्न–परिषुद्धता वाले परीक्षणों की कुछ अनुक्रियाओं के आरोपण की आवष्यकता होती है जब दो स्रोत पंक्तिबद्ध हों। इस कठिनाई को कम करने के लिए नीड़ित लाम्बिक एल एच डी को परिभाषित किया गया है।

लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाओं की संरचनपा की एक सामान्य पद्धति का वर्णन किया गया है। संरचना पद्धति का सरोकार प्रथम क्रम एवं द्विजरय क्रम लाम्बिक लैटिन हाइपरक्यूब में सुधार कर के लाम्बिक एवं लगभग लाम्बिक स्थान भरने ववाली लैटिन हाइपरक्यूब अभिकल्पनाओं की संरचना की गई है। नीडित लाम्बिक लैटिन हाइपरक्यमब (एनओएलएच) अभिकल्पनाओं की एक संरचना पद्धति का वर्णन किया गया है। नीडित लाम्बि हाइपरक्यूब अभिकल्पनाओं की संरचना करने वाली दो सामान्य पद्धतियां विकसित की गई है। प्रथम पद्धति का सरोकार एन ओ एल एच की 2 परतों से है और द्वितीय पद्धति का सरोकार एन ओ एल एच की तीन अथवा अधिक परतों से है। इन पद्धतियों से वर्तमान नीड़ित लाम्बिक लैटिन हाइपरकूयब अभिकल्पनाओं की अपेक्षा कम रनों की संख्या द्वारा बहुत सी नई नीड़ित लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाएं मिलती हैं।

लेखक, भा.कृ.अनु.प.–भा.कृ.सा.अनु.सं. निदशक को अनुसंधान कार्य को सफलतापूर्वक करने हेतु एवं समय समय पर सभी आवष्यक सुविधाएं प्रदान करने के लिए हार्दिक धन्यवाद अभिव्यक्त करते हैं। भा.कृ.अनु.प. –भा.कृ.सा.अनु.सं की परीक्षण अभिकल्पना प्रभाग की प्रभागाध्यक्षा एवं प्रभाग के अन्य वैज्ञानिकों से प्राप्त सहयोग के लिए धन्यवाद सहित आभार व्यक्त करते हैं। परीक्षण अभिकल्पना प्रभाग के भी देवेन्द्र कुमार एवं श्रीमति सुनिता द्वारा प्राप्त मदद के लिए आभारी हूं।

सुकांत दाश **राजेन्द्र प्रसाद** **बी.एन. मण्डल** **सुशील कुमार सरकार**

PREFACE

Latin hypercube designs is a popular choice of experimental design when computer simulation is used for studying a physical process as the design points of a Latin hypercube are equally spaced in the design region when projected onto univariate margins. In computer experiments, changing the levels of variables is only a matter of setting different numbers for the input, whereas in physical experiments, taking more levels of variables often requires an additional cost of making prototypes and a more elaborate and time-consuming implementation of the experiment. Therefore, the differences between computer experiments and traditional physical experiments call for different considerations in design and analysis methods for computer experiments. To handle such type of experimental situations Orthogonal Latin Hypercube design was introduced. Orthogonal Array (OA) designs are used extensively for planning experiments and their success is due to the uniformity properties but when a large number of factors are to be studied in an experiment and only a few of them are virtually effective, OA designs projected onto the subspace spanned by the effective factors can result in repetition of points on effective part only which is undesirable for physical experiments in which the bias of the proposed model is more serious than the variance. In this case LHD may be preferred. But the projection of such design points onto even bivariate margins cannot be guaranteed to be uniformly scattered. Thus to handle this situation Orthogonal arrays based Latin hypercube has been proposed which generally have better space filling properties than random Latin hypercube designs. In some experiments large and expensive computer code can be executed at various degrees of fidelity, and result in computer experiments with multiple levels of cost and accuracy. Efficient data collection from these experiments is critical. Nested designs are useful for designing such experiments. The main drawback of this approach to use LHD is that it requires imputation of some responses of the high-accuracy and low-accuracy experiments when the two sources are aligned together. To mitigate this difficulty, Nested orthogonal LHD has been defined.

A general method of construction of Orthogonal Latin Hypercube Designs has been describe. The methods of construction deal with both first order and second order orthogonal Latin hypercube designs. Orthogonal and nearly orthogonal space filling Latin Hypercube Designs has been constructed by modifying the OLH designs. A construction methods of Nested Orthogonal Latin Hypercube (NOLH) Designs has been described. Two general methods of constructing nested orthogonal Latin hypercube designs have been developed. First method deals with 2 layers of NOLH and the second methods deals with three or more layers of NOLH. The methods give many new nested orthogonal Latin hypercube designs with fewer number of runs as compared to existing nested orthogonal Latin hypercube designs.

The authors express their sincere thanks to the Director, ICAR-IASRI for providing all necessary facilities to carry out the research work successfully. The cooperation received from Head of Division and other scientists from the Division of Design of Experiments, ICAR-IASRI, New Delhi is duely acknowledged. The help rendered by Mr. Devender Kumar and Mrs. Sunita, Division of Design of Experiments is also duly acknowledged.

Sukanta Dash **Rajender Parsad** **Baidyanath Mandal** **Susheel Kumar Sarkar**

January 2019

विषय

Contents

CHAPTER I

INTRODUCTION AND REVIEW OF LITERATURE

1.1 Introduction

In scientific and engineering research, physical experimentation is often very expensive and time consuming. Because many physical systems can be described by mathematical equations, scientists are able to find numerical solutions of those equations to simulate the systems. Such applications have increased rapidly in the past decade. In contrast to a physical experiment, this kind of experiment is called a computer experiment, and the corresponding computer program a computer model. Given an input vector x, the response $f(x)$ of a computer model is often deterministic, so that considerations such as blocking and randomization for physical experiments are irrelevant. Also, in computer experiments, changing the levels of variables is only a matter of setting different numbers for the input, whereas in physical experiments, taking more levels of variables often requires an additional cost of making prototypes and a more elaborate and time-consuming implementation of the experiment. Therefore the differences between computer experiments and traditional physical experiments call for different considerations in design and analysis methods for computer experiments.

Latin hypercube designs introduced by McKay *et al*. (1979) have proved to be a popular choice of experimental design when computer simulation is used for studying a physical process as the design points of a Latin hypercube are equally spaced in the design region when projected onto univariate margins. A Latin hypercube design, LH(n, m), is an $n \times m$ matrix whose columns are permutations of the column vector (1, 2, ..., n). Each Latin hypercube has the property of achieving uniformity in each of the m univariate margins. In scientific and engineering research, physical experimentationis often very expensive and time consuming. Becausemany physical systems can be described by mathematicalequations, scientists are able to find numerical solutionsof those equations to simulate the systems. For example, injection molding are very much useful in cooling system (Tang, 1993) which are used in food processing. In an injection molding process, hot liquid material is injected into a mold and then forms to the desired space. To cool down the hot molding material, cooling systems are often used in injection molding processes. A good cooling system should reduce the material temperature quickly and uniformly. Because it is very difficult to measure the material temperature and to build prototypes of cooling systems, physical experiments would be very costly. Suchapplications have increased rapidly in the past decade. Incontrast to a physical experiment, this kind of experiment is called a computer experiment. In computer experiments, changing the levels of variables is only a matter of setting different numbers for the input, whereas in physical experiments, taking more levels of variables often requires an additional cost of making prototypes and a more elaborate and time-consuming implementation of the experiment. Therefore, the differences between computer experiments and traditional physical experiments call for different considerations in design and analysis methods for computer experiments. Orthogonal Array (OA) designs are used extensively for planning experiments and their success is due to the uniformity properties but when a large number of factors are to be studied in an experiment and only a few of them are virtually effective, OA designs projected onto the subspace spanned by the effective factors can result in repetition of points on effective part only which is undesirable for physical experiments in which the bias of the proposed model is more serious than the variance. In this case LHD may be preferred. But the projection of such design points onto even bivariate margins cannot be guaranteed to be uniformly scattered. Thus to handle this situation Tang(1993) proposed Orthogonal arrays based Latin hypercube which generally have better space filling properties than random Latin hypercube designs. In some experiments large and expensive computer code can be executed at various degrees of fidelity, and result in computer experiments with multiple levels of cost and accuracy. Efficient data collection from these experiments is critical. Nested designs, proposed by Qian and Wu (2008), are useful for designing such experiments. For drawing two independent and identically distributed samples, one with m runs chosen as D_1 and another with n runs chosen as D_2 , $m < n$. one can produce a Latin hypercube design of m runs and another Latin hypercube design with n runs as D_1 and D_2 , respectively. The main drawback of this approach to use LHD is that it requires imputation of some responses of the high-accuracy and low-accuracy experiments when the two sources are aligned together. To mitigate this difficulty, Qian (2009) introduced nested Latin hypercube designs. A nested Latin hypercube design with two layers is defined to be a special Latin hypercube design that contains a smaller Latin hypercube design as a subset, where the whole set is the first layer and the embedded smaller Latin hypercube design is the second layer. When such a design is used, the first layer is chosen as D_l and the second layer as D_2. Recently, Sun *et al*. (2013) presented a general approach to constructing nested spacefilling designs using nested difference matrices. Li and Qian (2013) suggested a method of construction of nested (nearly) orthogonal designs intended for multi-fidelity computer experiments. Such designs are two (nearly) orthogonal designs with one nested within the other. These nested space-filling designs (Sun *et al*., 2013) can achieve uniformity in low dimensions, but they lack orthogonality. Thus to overcome this situations Yang *et al*. (2014) proposed nested Orthogonal Latin hypercube design.

1.2 Review of Orthogonal LatinHypercube Designs

Latin hypercube designs were introduced by McKay *et al.* (1979) and proved to be a popular choice for experiments run on computer simulators and in global sensitivity analysis. They introduced theory behind the use of Latin hypercube design in computer experiments for estimation of expected output from a deterministic mathematical model by applying a particular distribution of inputs. Since their contribution seems to be very interesting then whole efforts are converted to find an easy construction method of a good Latin hypercube design.

Tang (1993) proposed orthogonal arrays based Latin hypercube which generally have better space filling properties than random Latin hypercube designs. He used orthogonal array (OA) for construction of Latin hypercube design and proved that when used for integration, a sampling scheme with OA-based Latin hypercube designs are more efficient than Latin hypercube sampling.

Tang (1994) obtained a method of construction of maximin Latin Hyper cube designs. The maximin Latin hypercube designs lead to maximization of minimum distances among the pairs of design points.

Ye (1998) proposed a class of orthogonal Latin hypercubes that preserves orthogonality among columns. He gave the method of construction of orthogonal Latin hypercube design for $n = 2^k$ or 2^{k+1} and $m = 2^{k-2}$ for any integer $k>1$. This method provides the estimation of all linear model parameter independently and provide nonparametric fitting procedure.

Steinberg and Lin (2006) provided a method of construction of orthogonal Latin hypercube designs for particular 2^{2^k} run sizes for any integer *k*. This method was obtained by rotating groups of factors in a two level 2^{2^k} run regular fractional design.

Cioppa and Lucas (2007) presented an algorithm for constructing orthogonal Latin hypercubes, given a fixed sample size,in more dimensions. They also gave a method of construction that improves the space-filling properties of the resultant Latin hypercubes at the expense of inducing small correlations between the columns in the design matrix. They gave OLH design for $n = 2^{k+1};\ m = k+1+\binom{k}{2}$ and $n = 2^{k+1}+1;\ m = k+1+\binom{k}{2}$, $k \le 11$.

Bingham *et al.* (2009) introduced a method for construction of orthogonal Latin hypercube designs by using two level fractional factorial designs. This method possess some more flexibility with respect to run and factor combination compare to method given by Ye (1998) and Steinberg and Lin(2006).

Pang *et al.* (2009) developed a method of constructionof orthogonal Latin Hypercubes of P^{2^k} runs and up to (P^{2^k} − 1)/(p − 1)factors by rotating groups of factors in a p-level P^{2^k} runs regular fractional designs, where p is a prime.

Lin *et al.*(2009) introduced a method of obtaining orthogonal or nearly orthogonal Latin hypercube by coupling an orthogonal array of index unity with a small orthogonal or nearly orthogonal Latin hypercube. They presented collection of some orthogonal or nearly orthogonal design for large factor run ratio and offer much more economical than existing methods.

Sun *et al.* (2009) introduced second-order orthogonal Latin hypercube for $n = 2^{k+1};\ m = 2^k$ and $n = 2^{k+1}+1;\ m = 2^k$; for any integer $k \ge 1$. Resulted designs from this method generate uniform samples for the marginal distribution of each input variable.

Lin*et al.* (2010) devloped a method for construction of orthogonal Latin hypercube design and nearly orthogonal design which is more flexible compare to existing methods with respect to factor run combination. They proved that no orthogonal Latin hypercube design exists for $4r + 2$ runs, where r any integer.

Sun *et al.* (2010) obtained a method of construction of orthogonal Latin hypercube designs for $n = r2^{k+1};\ m = 2^k$ and $n = r2^{k+1}+1;\ m = 2^k$ for all r integer and k<11.They improved the method of construction given by Sun *et al*. (2009).

Dey and Sarkar (2014) extended the result of Lin et al. (2009) on the construction of a large orthogonal Latin hypercube design by combining smaller orthogonal Latin hypercube design with an orthogonal array of strength two and reported several new 1st order OLH(*n*, *m*) design. Also reported some new 2nd order OLH (*n, m)* design by computer algorithm.

Parui (2015) developed method of construction of orthogonal Latin hypercube designs for 2 and 3 factors and any number of runs. He also developed method of construction of Latin Hypercube designs with good space filling property.

1.3 Review of Spacefilling Latin Hypercube Designs

Several researchers have contributed towards finding methods of construction of Latin hypercube design which gives good space-filling. Tang (1993) used orthogonal array (OA) for construction of Latin hypercube design and proved that when used for integration, a sampling scheme with OA-based Latin hypercube designs are more efficient than Latin hypercube sampling.

Tang (1994) obtained a method of construction of maximin Latin Hyper cube designs. The maximin Latin hypercube designs lead to maximization of minimum distances among the pairs of design points.

Morris and Mitchell (1995) developed an algorithm to find optimal Latin hypercube designs. Since, generally computer experiments involves large numbers of controlled variables then maximin Latin hypercube designs with moderate size are most preferable in the field of cuboidal design region where each and every input is represented by two level. On the basis of this, they mainly concentrated to find maximin Latin hypercube designs i.e. maximization of minimum distance between design points by using simulated annealing algorithm which will have good entropy criteria as well as good projective properties.

Ye *et al*. (2000) proposed an algorithm to find symmetric Latin hypercube designs by using exchange algorithm. They concentrated mainly on entropy criteria and the minimum intrinsic distance criteria as optimality criterion.

Jin *et al*. (2005) developed an algorithm to find optimal Latin hypercube designs based on optimality criterions (for example maximin distance criterion, entropy criterions and central L_2 discrepancy criterions) which reduces computational time. The proposed algorithm was based on global optimal search algorithm named Enhanced Stochastic Evolutionary (ESE) and was flexible for certain number of factor and runs combination.

Liefvendahl and Stocki (2005) compared the efficiency between columnwise-pairwise (CP) algorithm and genetic algorithm for the purpose of statistical investigations. They came to conclude that columnwise-pairwise algorithm is preferred for small Latin hypercube over genetic algorithm where genetic algorithm is preferred over columnwise-pairwise in case of large Latin hypercube designs.

Dam *et al.* (2007) finding maximin Latin hypercube design emphasizing mainly on two dimensions viz. two factors. They described two dimensional Latin hypercube design as non-attacking position of rooks on an $n \times n$ chessboards in such a way that minimal distance between the pair rooks maximized. They obtained a method of construction of Latin hypercube design for run size ≤ 70.

Viana *et al*. (2010) developed an algorithm to find optimal Latin hypercube design mainly emphasizing on minimization of Φp criterion i.e. a criterion based on minimization of maximum distance between design points. Their algorithm based on Transitional Propagation resulting minimal computational efforts and produces results virtually in real time. They compared their proposed algorithms with existing algorithms such as random search, genetic algorithm, enhanced stochastic evolutionary algorithm and concluded that the Φp criteria value tends to decrease as the dimension of Latin hypercube design increases from their algorithm. Further, for finding near optimal Latin hypercube designs up to medium dimension, their proposed Transitional Propagation algorithm possesses computational advantages.

Zhu *et al.* (2011) obtained an algorithm for finding maximin Latin hypercube design using successive local enumeration. They compared with existing algorithm namely *lhsdesign* function of MATLAB, binary coded genetic algorithm, permutation coded genetic algorithm and translation propagation algorithm and concluded that the algorithm proposed by them using successive local enumeration have good space-filling property as well as good projective property as compared to other algorithms.

Pan *et al.* (2014) developed translational propagation and successive local enumeration algorithm (TPSLE) to find optimal or near optimal Latin hypercube design by combing existing two algorithms viz. (i), translational propagation algorithm (Viana *et al.*, 2010) and (ii) successive local enumeration algorithm (Zhu *et al.*, 2011). They compared this algorithm with existing two algorithms and showed that TPSLE is efficient with respect to computational time, space-filling and projective property.

Latin hypercube designs with good space-filling properties can also be obtained using JMP version 10. JMP gives criteria values for a generated Latin hypercube design.

A review of literature on the construction procedure for finding a good space-filling Latin hypercube design is mainly based on computer aided search or algorithm. Generally algorithm based construction procedure fails to give unique result after repeated run and time consuming. Although some efforts have been done for finding a theoretical method for obtaining good Latin hypercube design, much more need to be done.

1.4 Review of Nested Orthogonal Latin Hypercube Designs

Qian and Wu (2008) proposed the nested designs, useful for designing such experiments and develop a method for nested Latin hypercube design. They also proposed some Bayesian hierarchical Gaussian process models for such type of computer experiments where nested designs are useful.

Qian *et al*.(2009a) proposed new types of designs called nested space-filling designs for conducting multiple computer experiments with different levels of accuracy. They developed several approaches to constructing such designs. They proposed a new approach of nested Latin hypercube designs where a nested Latin hypercube design with two layers is defined to be a special Latin hypercube design that contains a smaller Latin hypercube design as a subset, where the whole set is the first layer and the embedded smaller Latin hypercube design is the second layer.

Qian *et al*.(2009b) proposed a method for construction of some nested space-filling designs for computer experiments with two levels of accuracy. For the construction of nested space filling design the used Galois field and orthogonal array.

Yang *et al*. (2014) Proposed methods to construct nested Latin hypercube designs with two or more layers, by use of orthogonal designs. The constructed designs possess the propertythat the sum of the elementwise products of any three columns is zero. They introcuced the concept of nested orthogonal latin hypercube design.

1.5 Motivation

Latin hypercube designs introduced by McKay *et al*. (1979) have proved to be a popular choice of experimental design when computer simulation is used for studying a physical process as the design points of a Latin hypercube are equally spaced in the design region when projected onto univariate margins. A Latin hypercube design, LH(n, m), is an $n \times m$ matrix whose columns are permutations of the column vector (1, 2, ..., n). Each Latin hypercube has the property of achieving uniformity in each of the m univariate margins. In scientific and engineering research, physical experimentationis often very expensive and time consuming. Because many physical systems can be described by mathematical equations, scientists are able to find numerical solutions of those equations to simulate the systems. For example, injection molding are very much useful in cooling system (Tang, 1993) which are used in food processing. In an injection molding process, hot liquid material is injected into a mold and then forms to the desired space. To cool down the hot molding material, cooling systems are often used in injection molding processes. A good cooling system should reduce the material temperature quickly and uniformly. Because it is very difficult to measure the material temperature and to build prototypes of cooling systems, physical experiments would be very costly. Such applications have increased rapidly in the past decade. In contrast to a physical experiment, this kind of experiment is called a computer experiment. In computer experiments, changing the levels of variables is only a matter of setting different numbers for the input, whereas in physical experiments, taking more levels of variables often requires an additional cost of making prototypes and a more elaborate and time-consuming implementation of the experiment. Therefore, the differences between computer experiments and traditional physical experiments call for different considerations in design and analysis methods for computer experiments. To handle such type of experimental situations Orthogonal Latin Hypercube design was introduced by Tang (1993). Orthogonal Array (OA) designs are used extensively for planning experiments and their success is due to the uniformity properties but when a large number of factors are to be studied in an experiment and only a few of them are virtually effective, OA designs projected onto the subspace spanned by the effective factors can result in repetition of points on effective part only which is undesirable for physical experiments in which the bias of the proposed model is more serious than the variance. In this case LHD may be preferred. But the projection of such design points onto even bivariate margins cannot be guaranteed to be uniformly scattered. Thus to handle this situation Tang(1993) proposed Orthogonal arrays based Latin hypercube which generally have better space filling properties than random Latin hypercube designs. In some experiments large and expensive computer code can be executed at various degrees of fidelity, and result in computer experiments with multiple levels of cost and accuracy. Efficient data collection from these experiments is critical. Nested designs, proposed by Qian and Wu (2008), are useful for designing such experiments. For drawing two independent and identically distributed samples, one with m runs chosen as D_1 and another with n runs chosen as D_2 , $m < n$. one can produce a Latin hypercube design of m runs and another Latin hypercube design with n runs as D_1 and D_2 , respectively. The main drawback of this approach to use LHD is that it requires imputation of some responses of the high-accuracy and low-accuracy experiments when the two sources are aligned together. To mitigate this difficulty, Qian (2009) introduced nested Latin hypercube designs. A nested Latin hypercube design with two layers is defined to be a special Latin hypercube design that contains a smaller Latin hypercube design as a subset, where the whole set is the first layer and the embedded smaller Latin hypercube design is the second layer. When such a design is used, the first layer is chosen as D_l and the second layer as D_2. Recently, Sun et al. (2013) presented a general approach to constructing nested spacefilling designs using nested difference matrices. Li and Qian (2013) suggested a method of construction of nested (nearly) orthogonal designs intended for multi-fidelity computer experiments. Such designs are two (nearly) orthogonal designs with one nested within the other. These nested space-filling designs (Sun *et al*., 2013) can achieve uniformity in low dimensions, but they lack orthogonality. Thus to overcome this situations Yang *et al.* (2014) proposed nested Orthogonal Latin hypercube design. Keeping in view the above, the following objectives have been framed.

1.6 Objectives

i. To develop method(s) of construction for obtaining first order and second order orthogonal Latin hypercube designs.
ii. To develop method(s) of construction of orthogonal and/or nearly orthogonal Latin hypercube designs with good space filling property.
iii. To develop method(s) of construction of nested orthogonal Latin hypercube designs.
iv. To develop a web application for obtaining orthogonal and nested orthogonal Latin hypercube designs constructed from above methods of construction

1.7 Scope of the present study

The results obtained in the present investigation have been presented in Chapter II and Chapter III and chapter IV. In the Chapter II, a general method of construction of Orthogonal Latin Hypercube Designs has been describe. The methods of construction deal with both first order and second order orthogonal Latin hypercube designs. A Catalogue of Orthogonal Latin hypercube

Designs of 1st order and 2nd order with m (≤ 6) factors and n (≤ 20) runs is also presented in Appendix I and Appendix II respectively. The 1st order OLH(n, m) designs ensure independence of estimates of linear effects when a first order model is fitted whereas 2nd order OLH design ensures that not only the estimates of linear effects are mutually uncorrelated but they are also are uncorrelated with the estimates of quadratic and interaction effects in a second order model. A web application of generation of these designs has also been developed and presented in Chapter 2.

In the Chapter III, construction methods of Orthogonal and nearly orthogonal space filling Latin Hypercube Designs has been described. A Catalogue of Orthogonal and nearly orthogonal space filling Latin Hypercube Designs with m (≤ 6) factors and n (≤ 20) runs is presented in Appendix III(a). The orthogonality value and space filling values of the designs presented in Appendix III(a) is presented in Appendix III(b). Space-filling criterion provides maximum coverage to the whole design space and orthogonality criterion helps to estimate linear as well as higher order polynomial effects independently. To determine a Latin hypercube design with respect to space-filling property, three criterion like entropy criterion, $Ø_p$ criterion and central L_2 discrepancy criterion was used. A web application of generation of these designs has also been developed and presented in Chapter 3.

In the Chapter IV, construction methods of Nested Orthogonal Latin Hypercube (NOLH) Designs has been described. Two general methods of constructing nested orthogonal Latin hypercube designs have been developed. First method deals with 2 layers of NOLH and the second methods deals with three or more layers of NOLH. The methods give many new nested orthogonal Latin hypercube designs with fewer number of runs as compared to existing nested orthogonal Latin hypercube designs. A Catalogue of Nested Orthogonal Latinhypercube Designs of p(≤ 4) layers with m (≤ 6) factors and n (< 100) runs is presented in Appendix IV. A web application of generation of these designs has also been developed and presented in Chapter 4.

CHAPTER II

ORTHOGONAL LATIN HYPERCUBE DESIGNS

2.1 Introduction

Computer experiments are probably the most effective approach to probe the complex real world systems. This is especially true when the corresponding physical experiments are costly. Latin hypercube designs (LHDs), proposed by Mckay (1979), have been widely adopted in the design of computer experiments with quantitative factors because they spread the design points uniformly in any one-dimensional projection. An LHD with n runs and m factors is denoted by a matrix $L(n,m)=l_1, \ldots, l_m$ where l_j, j is the jth factor, and each factor includes n uniformly spaced levels $\{1,2,\ldots,n\}$. Often it is convenient to present the levels of the factors in an LHD in its centered form. To be specific, the levels belong to the set $\{-(n-1)/2, -(n-3)/2, \ldots, -(n-2i+1)/2,, (n-3)/2, (n-1)/2\}$.

An LHD when represented in its centered form is called an orthogonal LHD if the inner product of any two distinct columns is zero. Henceforth, we denote an orthogonal Latin hypercube design with n runs and m factors as an OLH(n,m). As discussed in Sun *et al*. (2009), an OLH ensures that the parameter estimates of the first order polynomial model

$$y = \beta_0 + \sum_{j=1}^{m} \beta_j x_j + \varepsilon \tag{2.1}$$

are uncorrelated. Recently, considerable attention is being given to the parameter estimation of second order polynomial model

$$y = \beta_0 + \sum_{j=1}^{m} \beta_j x_j + \sum_{j=1}^{m} \beta_{jj} x_j^2 + \sum_{j=1}^{m-1} \sum_{j'=j+1}^{m} \beta_{jj'} x_j x_{j'} + \varepsilon \tag{2.2}$$

When model (2.2) is used along with an OLH, it is desirable that the OLH should ensure that there is no correlation between parameter estimates of first and second order effects. This requires that the OLH should have the additional properties that (a) the entry-wise square of each column is orthogonal to all columns in the design and (b) the entry-wise product of any two distinct columns is orthogonal to all columns in the design. Such an OLH is called as second order OLH. Clearly, a second order OLH is always an OLH but an OLH may not be a second order OLH. Henceforth, unless we mention the word “second order", an OLH will simply refer to first order OLH. Several workers have given a number methods of construction of OLHs for different n and m. The values of n and m for second order OLHs are given below:

1. $n = 2^{k+1}$, m=2k and $n = 2^{k+1}$ +1, m = 2k (Ye *et al*.,1998)
2. $n = 2^{k+1}$, $m=k+1+\binom{k}{2}$ and $n = 2^{k+1}$ +1, $m=k+1+\binom{k}{2}$, $k \leq 11$ (Cioppa & Lucas, 2007)
3. $n = 2^{k+1}$, $m = 2^k$ and $n = 2^{k+1}$ +1, $m = 2^k$ (Sun *et al*.,2009)
4. $n = r2^{k+1}$, $m = 2^k$ and $n = r2^{k+1}$ +1, $m = 2^k$ (Sun *et al*.2010)

where $k \geq 1$ is an integer.

Lin (2008) provided several interesting construction methods for OLHs. Georgiou and Efthimiou (2014) presented several new classes of OLHs for m =12, 16, 20 and 24 factors. Dey and Sarkar (2014) reported some more results on the construction of OLH including a few new second order OLHs. Parui *et al.(*2016) developed method of construction of second order OLHs for 3 factors with any permissible number of runs for which such designs exist. Mandal *et. al.* (2016) extended the construction of OLHs for four factors for any permissible number of runs for which such designs exist. They also presented several derived classes of OLHs for larger number of columns. Evangelaras, (2016) proved that second order OLH does not exist for m>2 columns with n runs when $n \equiv 4$ (mod 8).

From above review, it is clear that there are many run sizes for which a construction method of OLH does not exist, particularly for second order OLHs. For example, the above methods do not allow construction of second order OLHs for n=8,9 (mod 16) runs, except the method of parui et. al. (2016) which is applicable for m=3 only. In this article, we propose procedures to obtain OLH(n,6) for all permissible n for which an OLH exists. This completes the search of OLH up to six factors for any possible runs for which such designs exist. We also give two new series of second order OLH designs for six factors.

2.2 Construction of First Order Orthogonal Latin Hypercube Designs

A Latin hypercube design, LH(n, m), is an $n \times m$ matrix whose columns are permutations of the column vector (1, 2, …, n)'. We shall represent the levels of each factor, in its centered form. For a positive integer n, let g_n be an $n \times 1$ vector with its ith element equal to (i-(n+1)/2), $1 \leq i \leq n$, and G_n be the set of all permutations of g_n. We denote the set of all levels as L. A centered Latin hypercube design is an $n \times m$ matrix with columns from G_n . The columns of a Latin hypercube design represent the input factors and the rows, the experimental runs.

Lemma 1: There exists no OLH(n, m) if $n \equiv 2$ (mod 4).

Lemma 1 is due to Lin *et al*. (2009). Next, we note the following result.

Lemma 2: There exists no OLH(n, 6) for $n < 11$ runs.

To prove this, it is known that for m =6, no OLH exists for $n < 6$ and n = 6, 10 is covered by Lemma 1. The non-existence of OLH(n, 6) for n = 7, 8 and 9 can be easily verified by an exhaustive search in computer. Thus, to have an OLH(n, 6), we need $n \geq 11$ runs.

We shall call an $n \times m$ matrix A=(a_1 , a_2 , …, a_m) as orthogonal if (a) $a_i' a_j = 0$ for all $i \neq j$ =1, 2, …, m. We denote an orthogonal matrix as OM(n, m). We shall call the matrix A second order orthogonal if it has the additional properties that (a) $\left(a_i^2\right)' a_j = 0$ (b) $\left(a_j^2\right)' a_i = 0$ for all $i \neq j$ =1, 2, …, m. and (c) a_i o a_j o a_k = 0 for all $i \neq j \neq k$ = 1,2,…, m where o denote the Hadamard product. Now, we present a result for constructing a second order OM (n, 6) with n ≡ 0 (mod 16).

Lemma 3: Let $\pm a_i, \pm b_i, \pm c_i, \pm d_i, \pm e_i, \pm f_i, \pm g_i, \pm h_i$, i = 1,2,…, t be any 16t real numbers such that none of them is 0. Consider the matrix $D = (D'_1, .D'_2, \ldots, D'_t)'$ with

$$D_i = \begin{bmatrix} H_i \\ -H_i \end{bmatrix} \text{with } H_i = \begin{bmatrix} a_i & -b_i & -d_i & -c_i & -h_i & e_i \\ b_i & a_i & -c_i & d_i & -g_i & -f_i \\ c_i & -d_i & b_i & a_i & -f_i & g_i \\ d_i & c_i & a_i & -b_i & -e_i & -h_i \\ e_i & -f_i & -h_i & g_i & -d_i & -a_i \\ f_i & e_i & -g_i & -h_i & c_i & b_i \\ g_i & -h_i & f_i & -e_i & b_i & -c_i \\ h_i & g_i & e_i & f_i & a_i & d_i \end{bmatrix} \qquad (2.3)$$

Then, the D is a second order OM (16t, 6) and each column of the matrix D contains each of the numbers $\pm a_i, \pm b_i, \pm c_i, \pm d_i, \pm e_i, \pm f_i, \pm g_i, \pm h_i$, i = 1,2,…, t exactly once.

Lemma 3 is very useful to construct second order OM (16t, 6). We give an example of constructing a second order OM (16, 6).

Example 1: Let t = 1 in Lemma 3 and set a_1 = -15/2, b_1 = -13/2, c_1 = -11/2, d_1 = -9/2, e_1 = -7/2, f_1 = -5/2, g_1 = -3/2, h_1 = -1/2. Then the matrix

$$D = \frac{1}{2}\begin{bmatrix} -15 & 13 & 9 & 11 & 1 & -7 \\ -13 & -15 & 11 & -9 & 3 & 5 \\ -11 & 9 & -13 & -15 & 5 & -3 \\ -9 & -11 & -15 & 13 & 7 & 1 \\ -7 & 5 & 1 & -3 & -9 & 15 \\ -5 & -7 & 3 & 1 & -11 & -13 \\ -3 & 1 & -5 & 7 & -13 & 11 \\ -1 & -3 & -7 & -5 & -15 & -9 \\ 15 & -13 & -9 & -11 & -1 & 7 \\ 13 & 15 & -11 & 9 & -3 & -5 \\ 11 & -9 & 13 & 15 & -5 & 3 \\ 9 & 11 & 15 & -13 & -7 & -1 \\ 7 & -5 & -1 & 3 & 9 & -15 \\ 5 & 7 & -3 & -1 & 11 & 13 \\ 3 & -1 & 5 & -7 & 13 & -11 \\ 1 & 3 & 7 & 5 & 15 & 9 \end{bmatrix} \quad (2.4)$$

is an OM (16,6).

Let D be a second order OM (16t, 6) as in Lemma 3. If each of the columns of the matrix D contains each of the elements of L exactly once, then D is a second order OLH (16t, 6).

Note that an OM constructed by Lemma 3 may or may not be an OLH. With the proper choice of the elements $a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i$, i = 1,2,…,t in Lemma 2, a second order OLH (16t, 6) can be constructed. Note that the design given in (2.4) is a second order OLH (16, 6).

Lemma 4: Let D_1 and D_2 be two orthogonal matrices of order $n_1 \times m$ and $n_2 \times m$, respectively, such that each of the columns of the matrix $D = \begin{pmatrix} D_1 \\ D_2 \end{pmatrix}$ contains each of the elements of L exactly once. Then, D is an OLH design with $n_1 + n_2$ rows and m columns.

Lemma 4 is a very useful result which can be utilized to construct bigger OLH designs by combining two smaller orthogonal matrices. In fact, Lemma 3 has been used by Sun *et al*. (2009) and Yang and Liu (2012) to construct OLH designs for $s2^r$ runs by combining orthogonal matrices with 2^r rows s times. Now, we present methods for constructing OLH designs for $m = 6$ for all n for which an OLH design may exist.

A run size n can fall into one of these twelve cases for which an OLH (n, 6) can exist: $n \equiv s \pmod{16}$, s = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15. Since $n \equiv$ 2,6,10 and 14 (mod 16) are equivalent to $n \equiv 2 \pmod 4$, an OLH does not exist for $n \equiv 2,6,10$ and 14 (mod 16) for any m by Lemma 1.

Using methods of Sun *et al.*(2009), Sun *et al.*(2010) and Yang and Lui (2012), one can construct an OLH for $n \equiv 0,1$ and 8 (mod 16), but not for other n. For the sake of completeness, we discuss construction of OLH designs for six columns for all the cases.

Theorem 1: An OLH (n, 6) can always be constructed for $n \equiv s \pmod{16}$, $s = 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15$ such that $n \geq 11$.

Proof. By Lemma 2, an OLH (n, 6) does not exist for $n < 11$. We give the construction of OLHs by dividing the range of n into three sub-ranges.

(i) $11 \leq n \leq 15$: An OLH (n, 6) for $n = 11,12,13$ and 15 is given in Table 1(a), (b), (c) and (d), respectively. These four designs are due to Lin (2008).

(ii) $16 \leq n \leq 31$: Let $n = 16 + s$ with $s = 0,1,3,4,5,7,8,9,11,12,13,15$. For $s = 0$ and 1, an OLH (16, 6) and OLH (17, 6) can be obtained following the method of Ye (1998) or Sun *et al.*(2009) or Yang and Liu (2012) . For $s = 3,4,5$, an OLH (n, 6) design with n = 19,20,21 is given in Table 2(a), (b) and (c), respectively. These three designs are due to Lin (2008). For $s = 7$, an OLH (23, 6) can be obtained by $\begin{pmatrix} D_1 \\ D_2 \end{pmatrix}$ where D_1 and D_2 are obtained by multiplying the elements of OLH (11, 6) and OLH (12, 6) given in Table 5(a) and (b) by 2. For $s = 8$, an OLH (24, 6) is given in Table 10(d) which is due to Georgiou and Efthimiou (2014). For $s = 9$, an OLH (25, 6) can be obtained by adding a row of zeros to the OLH (24, 6) in Table 10(d). For s = 11,12,13,15, n = 27,28,29,31 and to construct an OLH (n, 6) for these n, arrange the levels in L in ascending order and partition the levels into two disjoint sets S_1 and S_2 such that S_1 contains s levels (s = 11,12,13,15,) from the center of the arrangement and S_2 contains rest of the 16 levels. From the s levels in S_1, an OLH (s, 6) is available in Table 1 (a) to (d). From the 16 levels in S_2, construct an OM (16, 6) following Lemma 2. Now, following Lemma 2, juxtaposing these two orthogonal matrices gives an OLH (n, 6), n = 27,28,29,31.

(iii) $n \geq 32$: Let $n = 16(r + 1) + s$, $r = 1,2,\ldots$; $s = 0,1,3,4,5,7,8,9,11,12,13,15$. Denote the n levels as in L and arrange them in ascending order. Partition the levels into two disjoint sets S_1 and S_2 such that S_1 contains $16 + s$ levels from the center of the arrangement and S_2 contains rest of the $16r$ levels. From the $16 + s$ levels in S_1, construct an OLH ($16 + s$, 6) as described in

(ii) above. From the 16r levels in S_2, construct an OM (16r, 6) following Lemma 3. Now, following Lemma 3, juxtapose the OLH (16 + s, 6) and the OM (16r, 6) to get an OLH (n, 6).

For illustration, we give examples of construction of one OLH from each of case (ii) and (iii) above.

Let n = 28. We partition the 28 levels in L into two disjoint subsets S_1 and S_2 where S_1 = {-11/2, -9/2, -7/2, -5/2, -3/2, -1/2, 1/2, 3/2, 5/2, 7/2, 9/2, 11/2} and S_2 = {-27/2, -25/2, …, -15/2, -13/2, 13/2, 15/2, … ,25/2, 27/2} of sizes 12 and 16, respectively. From the elements in S_1, we get an OLH (12, 6) as in Table 1(b). From the elements in S_2, let t = 1 in Lemma 2 and set a_1 = -27/2, b_1 = -25/2, c_1 = -23/2, d_1 = -21/2, e_1 = -19/2, f_1 = -17/2, g_1 = -15/2, h_1 = -13/2 . Then the matrix

$$D = \frac{1}{2}\begin{bmatrix} -27 & 25 & 21 & 23 & 13 & -19 \\ -25 & -27 & 23 & -21 & 15 & 17 \\ -23 & 21 & -25 & -27 & 17 & -15 \\ -21 & -23 & -27 & 25 & 19 & 13 \\ -19 & 17 & 13 & -15 & -21 & 27 \\ -17 & -19 & 15 & 13 & -23 & -25 \\ -15 & 13 & -17 & 19 & -25 & 23 \\ -13 & -15 & -19 & -17 & -27 & -21 \\ 27 & -25 & -21 & -23 & -13 & 19 \\ 25 & 27 & -23 & 21 & -15 & -17 \\ 23 & -21 & 25 & 27 & -17 & 15 \\ 21 & 23 & 27 & -25 & -19 & -13 \\ 19 & -17 & -13 & 15 & 21 & 27 \\ 17 & 19 & -15 & -13 & 23 & 25 \\ 15 & -13 & 17 & -19 & 25 & -23 \\ 13 & 15 & 19 & 17 & 27 & 21 \end{bmatrix} \tag{2.5}$$

is an OM (16, 6) . Juxtaposing D with the design in Table 1(b), we get an OLH (28, 6).

Let n = 51. Here, r = 2, s = 3 in (iii) of the proof above. Partition the 51 levels in L into two disjoint subsvets S_1 and S_2 such that S_1 = { -9, -8, … , -1, 0, 1, … , 8, 9} and S_2 = {-25, -24, … , -11, -10, 10, 11, … , 24, 25}. Let t = 2 and let a_1 = -25, b_1 = -24, c_1 = -23, d_1 = -22, e_1 = -21, f_1 = -20, g_1 = -19, h_1 = -18, a_2 = -17, b_2 = -16, c_2 = -15, $d_{2,}$ = -14, e_2 = -13, f_2 = -12, g_2 = -11, h_2 = -10, in Lemma 3. Then, we get the following OM (32, 6) from Lemma 3.

$$D = \begin{pmatrix} -25 & 24 & 22 & 23 & 18 & -21 \\ -24 & -25 & 23 & -22 & 19 & 20 \\ -23 & 22 & -24 & -25 & 20 & -19 \\ -22 & -23 & -25 & 24 & 21 & 18 \\ -21 & 20 & 18 & -19 & -22 & 25 \\ -20 & -21 & 19 & 18 & -23 & -24 \\ -19 & 18 & -20 & 21 & -24 & 23 \\ -18 & -19 & -21 & -20 & -25 & -22 \\ 25 & -24 & -22 & -23 & -18 & 21 \\ 24 & 25 & -23 & 22 & -19 & -20 \\ 23 & -22 & 24 & 25 & -20 & 19 \\ 22 & 23 & 25 & -24 & -21 & -18 \\ 21 & -20 & -18 & 19 & 22 & -25 \\ 20 & 21 & -19 & -18 & 23 & 24 \\ 19 & -18 & 20 & -21 & 24 & -23 \\ 18 & 19 & 21 & 20 & 25 & 22 \\ -17 & 16 & 14 & 15 & 10 & -13 \\ -16 & -17 & 15 & -14 & 11 & 12 \\ -15 & 14 & -16 & -17 & 12 & -11 \\ -14 & -15 & -17 & 16 & 13 & 10 \\ -13 & 12 & 10 & -11 & -14 & 17 \\ -12 & -13 & 11 & 10 & -15 & -16 \\ -11 & 10 & -12 & 13 & -16 & 15 \\ -10 & -11 & -13 & -12 & -17 & -14 \\ 17 & -16 & -14 & -15 & -10 & 13 \\ 16 & 17 & -15 & 14 & -11 & -12 \\ 15 & -14 & 16 & 17 & -12 & 11 \\ 14 & 15 & 17 & -16 & -13 & -10 \\ 13 & -12 & -10 & 11 & 14 & -17 \\ 12 & 13 & -11 & -10 & 15 & 16 \\ 11 & -10 & 12 & -13 & 16 & -15 \\ 10 & 11 & 13 & 12 & 17 & 14 \end{pmatrix} \qquad (2.6)$$

Now, juxtaposing D with the OLH (19, 6) in Table 2(a), we get an OLH (51, 6).

Table 1: Orthogonal Latin Hypercube Designs for Six Factors for $11 \leq n \leq 15$

(a): n =11

$$\begin{bmatrix} -5 & -4 & -5 & -5 & -3 & 0 \\ -4 & 2 & -1 & 3 & 4 & 5 \\ -3 & -2 & 4 & 5 & -4 & -2 \\ -2 & 3 & -3 & 4 & 1 & -4 \\ -1 & 4 & 2 & -4 & 3 & 2 \\ 0 & -5 & 5 & -2 & 5 & -3 \\ 1 & 5 & 3 & -3 & -5 & -1 \\ 2 & -1 & 1 & 1 & -2 & 3 \\ 3 & 0 & 0 & -1 & 0 & 1 \\ 4 & 1 & -4 & 0 & 2 & -5 \\ 5 & -3 & -2 & 2 & -1 & 4 \end{bmatrix}$$

(b) : n =12

$$\frac{1}{2}\begin{bmatrix} -11 & -11 & -3 & -11 & -7 & -7 \\ -9 & -5 & -5 & 11 & 9 & 1 \\ -7 & 9 & 11 & -9 & -1 & 3 \\ -5 & 1 & 1 & 1 & 1 & 11 \\ -3 & 5 & -1 & 3 & 11 & -9 \\ -1 & 11 & 5 & 7 & -5 & -3 \\ 1 & 3 & -11 & 5 & -11 & -5 \\ 3 & -3 & 3 & -3 & 3 & 5 \\ 5 & -9 & 7 & 9 & -9 & 7 \\ 7 & -1 & -9 & -7 & 7 & 9 \\ 9 & 7 & -7 & -5 & -3 & -1 \\ 11 & -7 & 9 & -1 & 5 & -11 \end{bmatrix}$$

(c) : *n* =13

$$\begin{bmatrix} -6 & -6 & -6 & 0 & -5 & -1 \\ -5 & 1 & 4 & -1 & 6 & 5 \\ -4 & 6 & -4 & 5 & 5 & -2 \\ -3 & 2 & 6 & -4 & -6 & 2 \\ -2 & -2 & 2 & 2 & -2 & -4 \\ -1 & 3 & 1 & 1 & -3 & 3 \\ 0 & 4 & -2 & -6 & 1 & -5 \\ 1 & -4 & -5 & -2 & 3 & 4 \\ 2 & -5 & 5 & 6 & 2 & 1 \\ 3 & -3 & 3 & -5 & 4 & -6 \\ 4 & -1 & 0 & 4 & -1 & -3 \\ 5 & 5 & -1 & 3 & -4 & 0 \\ 6 & 0 & -3 & -3 & 0 & 6 \end{bmatrix}$$

(d) : *n* =15

$$\begin{bmatrix} -7 & 4 & -7 & -6 & -2 & -5 \\ -6 & 3 & 5 & 3 & 5 & 2 \\ -5 & -6 & -2 & 5 & 6 & -3 \\ -4 & 1 & 4 & 4 & -5 & -2 \\ -3 & 0 & -4 & -7 & 0 & 3 \\ -2 & -2 & 3 & 0 & -4 & -7 \\ -1 & -7 & 6 & -2 & -3 & 5 \\ 0 & 7 & -5 & 7 & 3 & 7 \\ 1 & -4 & 1 & -5 & 4 & 6 \\ 2 & -1 & 0 & 6 & -6 & 1 \\ 3 & 6 & 2 & -3 & -7 & 4 \\ 4 & -3 & -3 & -1 & 2 & -1 \\ 5 & 5 & 7 & -4 & 7 & -4 \\ 6 & -5 & -6 & 1 & 1 & 0 \\ 7 & 2 & -1 & 2 & 1 & -6 \end{bmatrix}$$

Table 2: Orthogonal Latin Hypercube Designs for Six Factors for 19, 20, 21 and 24

(a) : *n* =19

$$\begin{bmatrix} -9 & -9 & -9 & -8 & 2 & 7 \\ -8 & 5 & 5 & 4 & -8 & 9 \\ -7 & -6 & -5 & 5 & 0 & -5 \\ -6 & -3 & 0 & -9 & -4 & -2 \\ -5 & 9 & 6 & 1 & 7 & 4 \\ -4 & 7 & 9 & -5 & -2 & -6 \\ -3 & -2 & 3 & 3 & -9 & -7 \\ -2 & 6 & -8 & 6 & 6 & -3 \\ -1 & -4 & 4 & 9 & 5 & 2 \\ 0 & 4 & -6 & -2 & 9 & - \\ 1 & 0 & -2 & -4 & -5 & -1 \\ 2 & 8 & -3 & -1 & 3 & 1 \\ 3 & -8 & 8 & 0 & 4 & 3 \\ 4 & -7 & 2 & 8 & -1 & -9 \\ 5 & -5 & 7 & -3 & 8 & 5 \\ 6 & 1 & -4 & 7 & -7 & 6 \\ 7 & 2 & -7 & 2 & -3 & 8 \\ 8 & 3 & -1 & -6 & -6 & -4 \\ 9 & -1 & 1 & -7 & 1 & 0 \end{bmatrix}$$

(b) : *n* =20

$$\frac{1}{2}\begin{bmatrix} -19 & 13 & -19 & -17 & -19 & 7 \\ -17 & -13 & 9 & 17 & -11 & -3 \\ -15 & -9 & 11 & 5 & 11 & 9 \\ -13 & 9 & -3 & -9 & 15 & 3 \\ -11 & 15 & 1 & -15 & -13 & -15 \\ -9 & 19 & -1 & 11 & 13 & -9 \\ -7 & 15 & -9 & 9 & -7 & 5 \\ -5 & -3 & 7 & -3 & 1 & 17 \\ -3 & -19 & -7 & 7 & 7 & 15 \\ -1 & 1 & 15 & -11 & -3 & -13 \\ 1 & -1 & 17 & -1 & -1 & -5 \\ 3 & -17 & -11 & 13 & 9 & -17 \\ 5 & 5 & -5 & -13 & 3 & -7 \\ 7 & 11 & 3 & 3 & 19 & -1 \\ 9 & 3 & -15 & 1 & 17 & 1 \\ 11 & 17 & 19 & 15 & -15 & -11 \\ 13 & -5 & -13 & 19 & -17 & 13 \\ 15 & -11 & 13 & -19 & 5 & 11 \\ 17 & -7 & -17 & -7 & -5 & -19 \\ 19 & 7 & 5 & -5 & -9 & 19 \end{bmatrix}$$

(c) : *n* =21

$$\begin{bmatrix} -10 & -2 & -9 & 4 & -9 & -9 \\ -9 & -5 & 8 & 9 & 10 & -7 \\ -8 & 0 & -3 & -8 & -4 & -6 \\ -7 & 10 & 4 & 7 & 6 & 6 \\ -6 & 9 & 3 & -5 & 4 & -5 \\ -5 & 5 & -1 & -1 & -7 & 3 \\ -4 & 1 & -2 & -9 & -5 & 9 \\ -3 & -4 & 7 & 2 & -1 & 10 \\ -2 & 6 & -5 & -4 & 8 & 8 \\ -1 & -7 & 9 & 6 & -10 & -1 \\ 0 & -3 & 0 & -6 & 3 & 0 \\ 1 & -1 & -6 & 10 & -3 & 5 \\ 2 & -8 & -4 & -2 & 2 & 7 \\ 3 & -10 & 5 & -3 & 5 & -8 \\ 4 & -6 & 2 & -10 & 7 & 1 \\ 5 & -9 & -7 & 0 & -2 & 2 \\ 6 & 2 & -10 & 5 & 9 & -4 \\ 7 & 8 & 1 & -7 & -6 & -10 \\ 8 & 4 & -8 & 8 & 1 & -3 \\ 9 & 3 & 6 & 3 & -8 & 4 \\ 10 & 7 & 10 & 1 & 0 & -2 \end{bmatrix}$$

(d) : *n* =24

$$\frac{1}{2}\begin{bmatrix} 15 & -5 & 19 & 23 & -21 & 17 \\ 19 & 15 & -5 & -21 & 17 & 23 \\ -5 & 19 & 15 & 17 & 23 & -21 \\ -23 & 21 & -17 & 15 & -5 & 19 \\ 21 & -17 & -23 & 19 & 15 & -5 \\ -17 & -23 & 21 & -5 & 19 & 15 \\ 7 & -3 & -1 & 11 & 13 & 9 \\ -3 & -1 & 7 & 13 & 9 & 11 \\ -1 & 7 & -3 & 9 & 11 & 13 \\ -13 & -11 & -9 & -3 & 7 & -1 \\ -11 & -9 & -13 & 7 & -1 & -3 \\ -9 & -13 & -11 & -1 & -3 & 7 \\ -15 & 5 & -19 & -23 & 21 & -17 \\ -19 & -15 & 5 & 21 & -17 & -23 \\ 5 & -19 & -15 & -17 & -23 & 21 \\ 23 & -21 & 17 & -15 & 5 & -19 \\ -21 & 17 & 23 & -19 & -15 & 5 \\ 17 & 23 & -21 & 5 & -19 & -15 \\ -7 & 3 & 1 & -11 & -13 & -9 \\ 3 & 1 & -7 & -13 & -9 & -11 \\ 1 & -7 & 3 & -9 & -11 & -13 \\ 13 & 11 & 9 & 3 & -7 & 1 \\ 11 & 9 & 13 & -7 & 1 & 3 \\ 9 & 13 & 11 & 1 & 3 & -7 \end{bmatrix}$$

Remark 1: Lemma 2 and Theorem 1 completely solves the existence and construction problem of OLHs for six factors. From the construction technique outlined in the proof of Theorem 1, one can get infinite number of new OLHs with six factors.

2.3 Construction of Second Order Orthogonal Latin Hypercube Designs

If the OLH constructed from the levels S_1 in the proof of Theorem 2 is of second order, then the final OLH design is also of second order.

Now, we have the following result.

Theorem 2: A second order OLH (n, 6) can always be constructed for $n \equiv s(\text{mod}16)$, s = 8, 9 with $n \geq 24$.

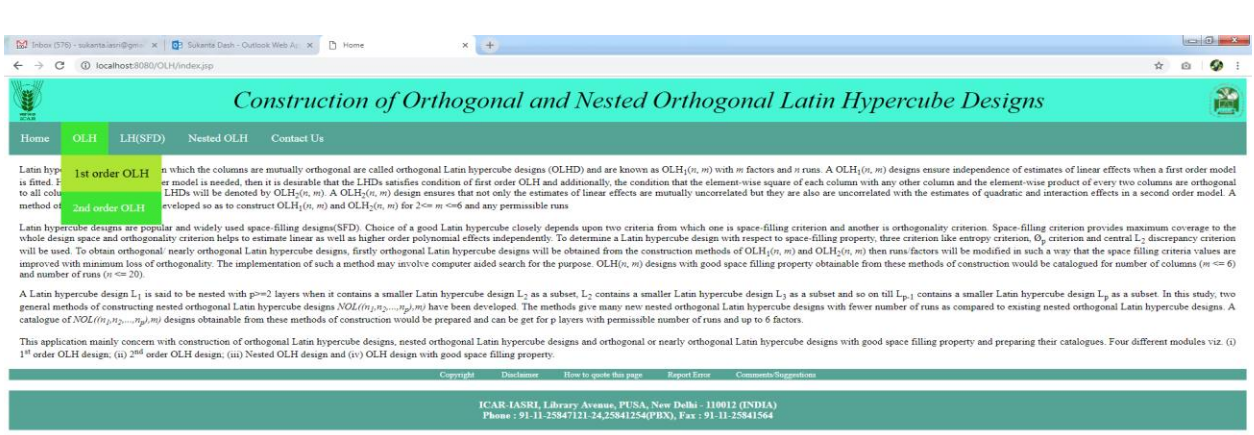
Proof. Let $n = 16(r+1) + s$, $r = 0,1,2, \ldots, s = 8,9,$. Then, arrange the n levels in ascending order and partition the levels into two disjoint sets S_1 and S_2 of sizes 16 + *s* and 16r respectively. For *s* = 8, there are 24 elements in S_1 and a second order OLH (24, 6) is readily available in Table 10(d). For *s* = 9, we can get a second order OLH (25, 6) by adding a row of zeros to the OLH (24, 6). From the 16r elements in the set S_2, a second order OM (16r, 6) can be constructed following Lemma 2. Juxtaposing the second order OM with the second order OLH, we get a second order OLH (16*r* + 16 + *s*,6).

Theorem 2 gives a new series of second order OLH designs for six factors. For example, available methods can not construct a second order OLH with 40 runs for 6 factors.

The method of construction given in Section 2.2 is general in nature and can be used for obtaining Orthogonal Latin Hypercube Designs. Using this method, a catalogue A Catalogue of Orthogonal Latin hypercube Designs of 1st order and 2nd order with *m* (≤ 6) factors and *n* (≤ 20) runs is also presented in Appendix I and Appendix II respectively.

2.4 Web Application

In this section, we describe a web application developed for online generation of Orthogonal Latin hypercube Designs of 1st order and 2nd order. The application has been developed using JSP language and STS (Java) platform. Some screen shots are given below:



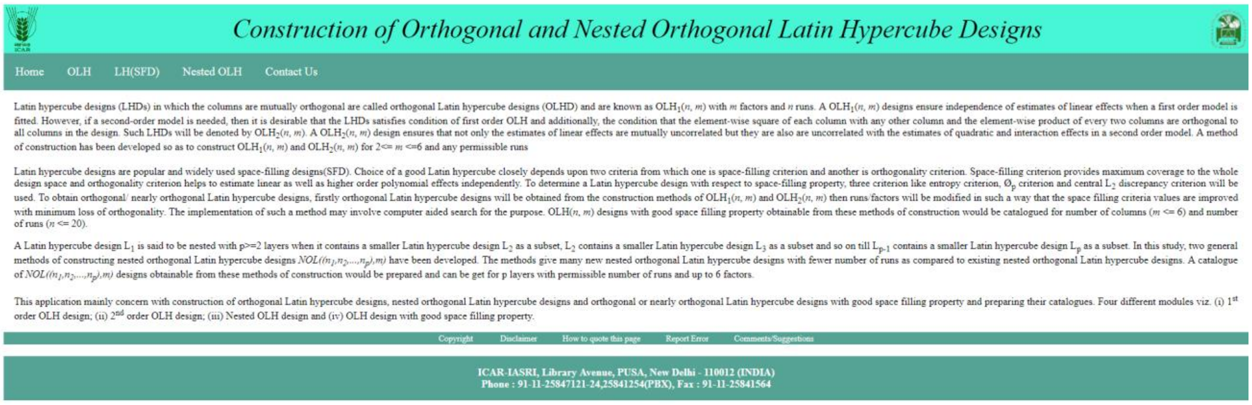
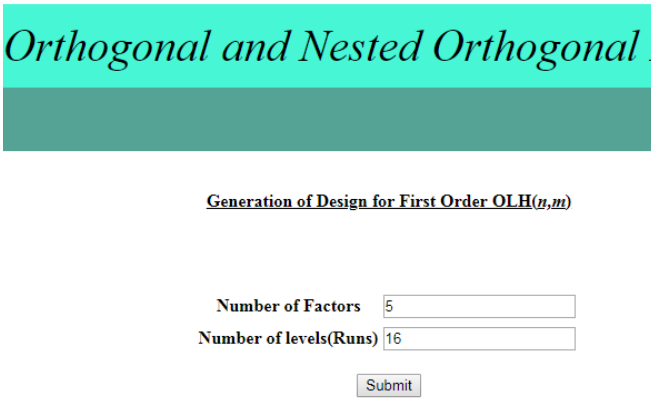


Figure 2.4.1: First Screen depicting Selection of number of factors

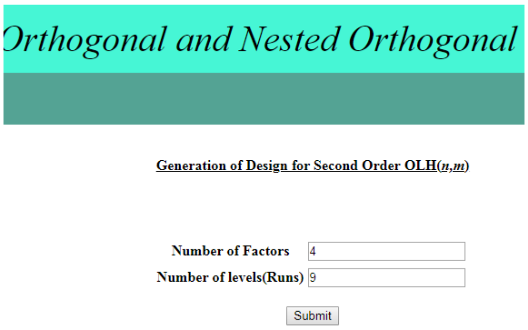


Number of levels (Runs) = 16 and Number of Factors = 5

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-7.5	6.5	4.5	5.5	0.5
Run 2	-6.5	-7.5	5.5	-4.5	1.5
Run 3	-5.5	4.5	-6.5	-7.5	2.5
Run 4	-4.5	-5.5	-7.5	6.5	3.5
Run 5	-3.5	2.5	0.5	-1.5	-4.5
Run 6	-2.5	-3.5	1.5	0.5	-5.5
Run 7	-1.5	0.5	-2.5	3.5	-6.5
Run 8	-0.5	-1.5	-3.5	-2.5	-7.5
Run 9	7.5	-6.5	-4.5	-5.5	-0.5
Run 10	6.5	7.5	-5.5	4.5	-1.5
Run 11	5.5	-4.5	6.5	7.5	-2.5
Run 12	4.5	5.5	7.5	-6.5	-3.5
Run 13	3.5	-2.5	-0.5	1.5	4.5
Run 14	2.5	3.5	-1.5	-0.5	5.5
Run 15	1.5	-0.5	2.5	-3.5	6.5
Run 16	0.5	1.5	3.5	2.5	7.5

Back

Save



Number of levels (Runs) = 9 and Number of Factors = 4

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-4	-3	-2	-1
Run 2	-3	4	-1	2
Run 3	-2	1	4	-3
Run 4	-1	-2	3	4
Run 5	4	3	2	1
Run 6	3	-4	1	-2
Run 7	2	-1	-4	3
Run 8	1	2	-3	-4
Run 9	0	0	0	0

Back

Save

2.5 Discussion

In the present investigation, we have completely solved the existence and construction problem of orthogonal Latin hypercube designs with six factors for any number of runs. To be specific, we have shown that an OLH(n, 6) does not exist for $n < 11$ and for $n \equiv 2 \pmod 4$, and also proved that an OLH(n, 6) always can be constructed for $n \equiv s(\text{mod}16)$, $s = 0,1,3,4,5,7,8,9,11,12,13,15$. We have also presented two new series of second order OLH for $n \equiv s(\text{mod}16)$,), $s = 8, 9$ for six factors.

CHAPTER III

ORTHOGONAL AND NEARLY ORTHOGONAL SPACE FILLING LATIN HYPERCUBE DESIGNS

3.1 Introduction

Latin hypercube designs are widely used space-filling designs. Choice of a good Latin hypercube closely depends upon two criteria viz. space-filling criteria and orthogonality. Space-filling criterion provides maximum coverage to the whole design space and orthogonality criterion helps to estimate parameters independently. As we know that criterion of orthogonality and space filling are not related to each other. An orthogonal Latin hypercube designs may not have good space-filling property and vice-versa.

3.2 Space filling Criterions for Latin Hypercube Designs

Three criteria (defined in Sections 3.2.1, 3.2.2 and 3.2.3) are being used widely for studying the space filling properties of a Latin hypercube/ Lain hypercube design. Measure of orthogonality of a Latin hypercube is defined in Section 3.5 and general approach used for constructing Latin hypercube designs is discussed in Section 3.6.

3.2.1 Entropy Criterion

Shanon (1948) introduced entropy criterion as a measure of 'amount of information' available from a design. An efficient design can be obtained by minimizing the mathematical form which was given by Shanon (1948). Later Koehler and Owen (1996) modified it to more simplified and analytical form for obtaining maximum entropy Latin hypercube design ($D_{n\times k}$) by minimizing following expression $-\log|\mathbf{R}|$, where R is correlation matrix.

$$R_{ij} = \sigma^2 \exp\left(-\theta\sum_{l=1}^{k}|s_{il} - t_{jl}|^q\right) \tag{3.1}$$

where s_{ij} and t_{jl} are two design points $\forall$ $1 \le i,\ j \le\ n$ and $1 \le q \le 2$, k is number of columns, θ is a constant and σ^2 is generally assumed to be 1. A Latin hypercube design with smaller entropy criterion value has better space-filling property.

3.2.2 Φ_p Criterion

Morris and Mitchell (1995) introduced the concept of maximin distance criterion and which is quite popular criterion for finding optimal Latin hypercube designs with good space filling properties. For given *n* and *k*, a Latin hypercube design with minimum Φ_p value is the best. It maximizes the minimum intrinsic distances between design points i.e. *max min* $\delta\left(X_i - X_j\right)$ for all $i \neq j,$ where X_i and X_j are corresponding i^{th} and j^{th} run of a Latin hypercube design. The Φ_p value can be directly calculated by using the following expression.

$$\Phi_p = \left[\sum_{i=1}^{s} J_i \delta_i^{-p} \right]^{\frac{1}{p}} \tag{3.2}$$

where J be the index i.e. ($J_1, J_2, \ldots, J_S$) is the number of time ($\delta_1, \delta_2, \ldots, \delta_S$) distances occur and $s \leq {}^nC_2$, p being any integer. A Latin hypercube design with lower Φ_p value is preferable in terms of space filling property.

3.2.3 Central L_2 Discrepancy Criterion

This criterion was introduced by Hikernell (1998). It measures the difference between empirical cumulative distribution function of a design and the uniform cumulative distribution function. Optimal Latin hypercube design can be obtained by minimizing the following mathematical form.

$$\begin{aligned}(CL_2)^2 &= \left(\frac{13}{12}\right)^k - \frac{2}{n}\sum_{i=1}^{n}\prod_{l=1}^{k}\left(1+\frac{1}{2}|d_{il}-0.5|-\frac{1}{2}|d_{il}-0.5|^2\right) \\ &+ \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\prod_{l=1}^{k}\left(1+\frac{1}{2}|d_{il}-0.5|+\frac{1}{2}|d_{jl}-0.5|-\frac{1}{2}|d_{il}-d_{jl}|\right)\end{aligned} \tag{3.3}$$

where d_{ij} are design points, k be the number of factors and n is the number of runs in a Latin hypercube design. The smaller the central L_2 discrepancy criterion value, the better a Latin hypercube design is in terms of space filling property.

Let $\mathbf{A} = \left(a_{ij}\right)$ be a Latin hypercube. Then a Latin hypercube design can be obtained by assuring all design points lie with in $[0, 1)^k$ is given below.

$$\mathbf{D} = \left[a_{ij} - \min(x_{ij})\right] / \left[\max(a_{ij}) - \min\left(a_{ij}\right)\right] \tag{3.4}$$

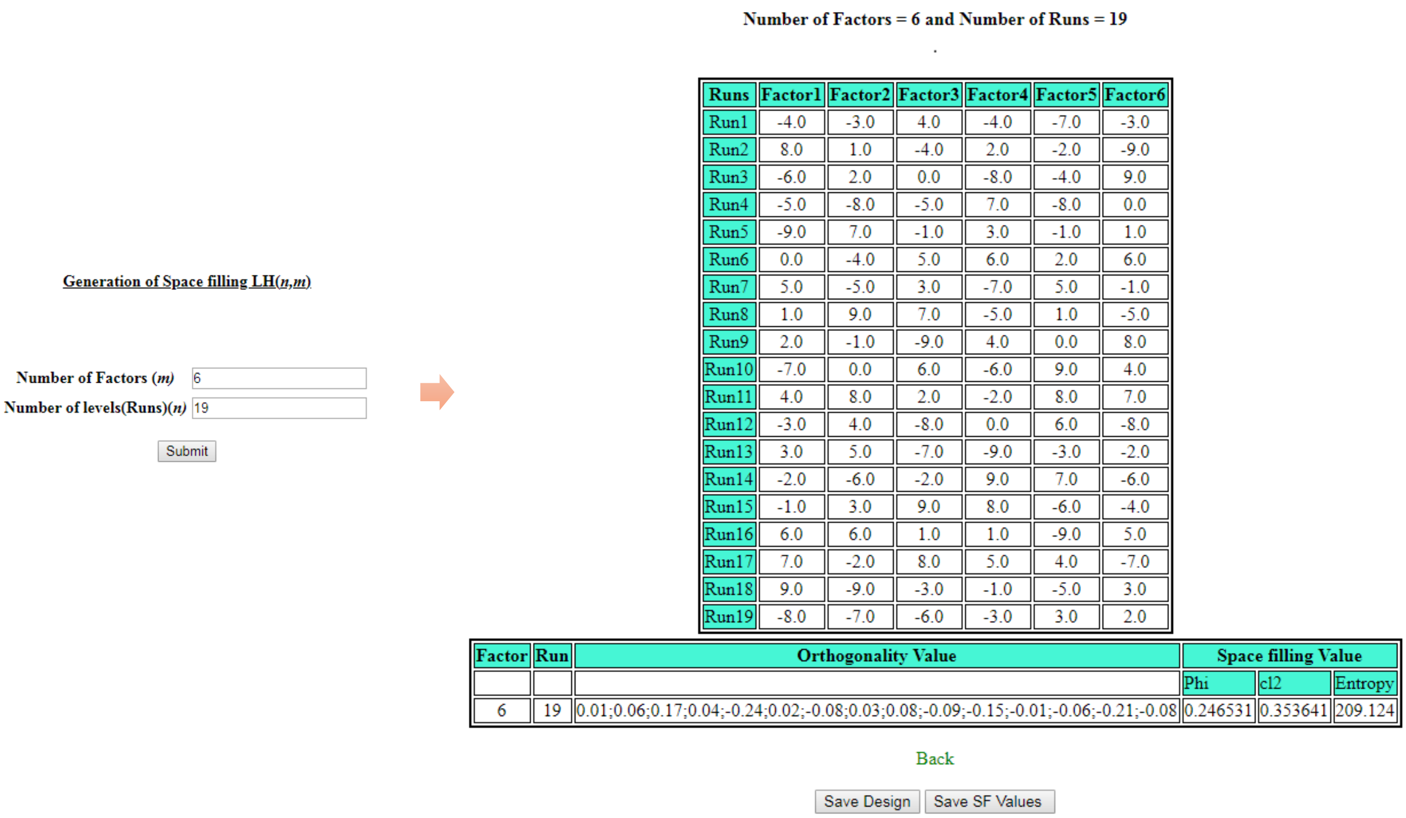
3.3 Construction of orthogonal and Nearly Orthogonal Space filling Latin Hypercube Designs

A Latin hypercube with minimum values of measures of space filling properties among all possible Latin hypercubes of given number of runs and factors is known as Optimal Latin hypercube with respect to space filling. Optimal design with respect to good space-filling values may also be obtained by minimizing simultaneously all these three criterion values.

Standard Latin squares of odd order and technique of rearrangement of elements in a specific order have been used for obtaining general methods of construction of Latin hypercube designs for two factors in any number of runs. Space-filling property and orthogonality of Latin hypercube obtained from proposed methods of construction are studied using the criteria defined in Sections 3.21, 3.2.2, and 3.2.3 . The techniques of augmenting base designs in smaller number of runs to obtain orthogonal Latin hypercube designs in larger run sizes and specific column arrangements have been used for obtaining general methods of construction for orthogonal Latin hypercube designs for upto six factors and 20 runs. Some modification in constructed OLH of 1st order designs have been used to obtain a orthogonal and nearly orthogonal Latin hypercube deisgns witth good space filling property. Space-filling property of orthogonal Latin hypercubes obtained from proposed methods of construction is studied using the criteria defined in Sections 3.21, 3.2.2, and 3.2.3 . PROC IML of SAS is used for developing codes for generating the designs obtainable from general methods of construction and to obtain values of three space filling criteria.

3.4 Web Application

In this section, we describe a web application developed for online generation of Orthogonal Latin hypercube Designs of 1st order and 2nd order. The application has been developed using JSP language and STS (Java) platform. Some screen shots are given below:



Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-4.0	-3.0	4.0	-4.0	-7.0	-3.0
Run2	8.0	1.0	-4.0	2.0	-2.0	-9.0
Run3	-6.0	2.0	0.0	-8.0	-4.0	9.0
Run4	-5.0	-8.0	-5.0	7.0	-8.0	0.0
Run5	-9.0	7.0	-1.0	3.0	-1.0	1.0
Run6	0.0	-4.0	5.0	6.0	2.0	6.0
Run7	5.0	-5.0	3.0	-7.0	5.0	-1.0
Run8	1.0	9.0	7.0	-5.0	1.0	-5.0
Run9	2.0	-1.0	-9.0	4.0	0.0	8.0
Run10	-7.0	0.0	6.0	-6.0	9.0	4.0
Run11	4.0	8.0	2.0	-2.0	8.0	7.0
Run12	-3.0	4.0	-8.0	0.0	6.0	-8.0
Run13	3.0	5.0	-7.0	-9.0	-3.0	-2.0
Run14	-2.0	-6.0	-2.0	9.0	7.0	-6.0
Run15	-1.0	3.0	9.0	8.0	-6.0	-4.0
Run16	6.0	6.0	1.0	1.0	-9.0	5.0
Run17	7.0	-2.0	8.0	5.0	4.0	-7.0
Run18	9.0	-9.0	-3.0	-1.0	-5.0	3.0
Run19	-8.0	-7.0	-6.0	-3.0	3.0	2.0

Factor	Run	Orthogonality Value	Space filling Value		
			Phi	cl2	Entropy
6	19	0.01;0.06;0.17;0.04;-0.24;0.02;-0.08;0.03;0.08;-0.09;-0.15;-0.01;-0.06;-0.21;-0.08	0.246531	0.353641	209.124

3.5 Discussion

Choice of a good Latin hypercube closely depends upon two criteria from which one is space-filling criterion and another is orthogonality criterion. Space-filling criterion provides maximum coverage to the whole design space and orthogonality criterion helps to estimate linear as well as higher order polynomial effects independently. In the present investigation, to determine a Latin hypercube design with respect to space-filling property, three criterion like entropy criterion, Φ_p criterion and central L_2 discrepancy criterion will be used. To obtain orthogonal/ nearly orthogonal Latin hypercube designs, firstly orthogonal Latin hypercube designs will be obtained from the construction methods of $OLH_1(n, m)$ and $OLH_2(n, m)$ then runs/factors will be modified in such a way that the space filling criteria values are improved with minimum loss of orthogonality. The implementation of such a method may involve computer aided search for the purpose. OLH(n, m) designs with good space filling property obtainable from these methods of construction would be catalogued for number of columns (m <= 6) and number of runs (n <= 20).

CHAPTER IV

NESTED ORTHOGONAL LATIN HYPERCUBE DESIGNS

4.1 Introduction

Computer experiments are now a days widely used to probe complex real world systems when the corresponding physical experiments are costly or impossible to perform. Latin hypercube designs (LHDs) introduced by McKay *et. al.* (1979) are widely used in the designing of computer experiments. An LHD with n runs and m factors is denoted by a matrix $L(n, m) = (l_1,\ldots,l_m,)$, where l_j, $j = 1, 2, \ldots$, m is the j^{th} factor, and each factor includes n uniformly spaced levels {1, 2,…, n}. An LHD is said to be orthogonal if the correlation coefficient between any two columns is zero. When represented in centered form, an orthogonal LHD has the property that the inner product of any two distinct columns is zero. Henceforth, we denote an orthogonal LHD with n runs and m factors as an OLH (n, m). As discussed in Sun *et al*. (2009), an OLH ensures that the parameter estimates of the first order polynomial model

$$y = b_0 + \sum_{j=1}^{m} b_j x_j + e \quad (4.1)$$

are uncorrelated. Construction of OLHs are studied by a number of authors, see (Ye, 1998; Cioppa and Lucas, 2007; Sun *et al*., 2009, 2010; Lin *et al*.,2010; Georgiou and Efthimiou, 2014; Dey and Sarkar, 2014; Parui *et al*.,2016; Mandal *et al*., 2016) and Sun and Tang (2017) among others.

Sometimes computer experiments may be performed with two or more different accuracy levels to study a complex physical system, see for example, Qian *et al*. (2006). Qian (2009) proposed nested Latin hypercube designs which are useful to conduct such type of experiments. A Latin hypercube design L_1 is said to be nested with $p \geq 2$ layers when it contains a smaller Latin hypercube design L_2 as a subset, L_2 contains a smaller Latin hypercube design L_3 as a subset and so on till L_{p-1} contains a smaller Latin hypercube design L_p as a subset. Clearly, $n_p < n_{p-1} < \ldots < n_1$. Further, if each L_i is an OLH, then L_1 is a nested OLH and is denoted as NOL((n_1, n_2, … , n_p), m). The p layers of the nested OLH are $L_1, L_2,\ldots, L_p$ respectively.

Recently, Yang *et al*. (2014) proposed methods of construction of nested OLH. They used a special type of $m \times m$ orthogonal design (OD) to construct nested OLH. Though their methods are quite appealing, the methods have two limitations. Firstly, the methods work for $m = 2^r$, r being a positive integer and secondly NOLs obtained through their proposed methods often have larger number of runs for a given *m*. In this article, we propose two very simple methods of construction of nested OLHs with *p* layers. The methods use existing OLHs to obtain a nested OLH. Using the proposed methods, we obtain many new nested OLHs with *p* layers.

4.2 Construction of nested OLHs with two layers:

In this section, we proposed a method of construction of nested OLHs with two layers. The construction depends on the existence of two OLH designs with specific number of runs. We shall represent the levels of each factor of the constituent OLHs in centered form, i.e., for OLH (*n*, *m*), the levels of each factor belong to the set {–(*n*–1)/2,–(*n*–3)/2,…,(*n*–3)/2,(*n*–1)/2}.

Theorem 1: *Let* L_1^* *be an OLH(n,m) and* L_2^* *be an OLH (n + 1, m) such that* $n \neq 1, 2 \bmod 4$. *Then* $L_1 = (2L_1^{*T}, L_2^{*T})^{T}$ *is both NOL ((2n + 1, n),m) and NOL((2n + 1, n+1), m).*

Proof. First we prove one-dimensional uniformity. Since L_1^* is an OLH (*n*, *m*), the elements in each column of L_1^* belong to the set {-(*n* - 1)/2, -(*n* – 3)/2, … ,-(*n* – 2i + 1)/2, … ,(*n* - 3)/2, (*n* – 1)/2}. Similarly, the elements in each column of L_2^* belong to the set {-*n*/2, -(*n* - 2)/2, … ,-(*n* – 2*i* + 2)/2,…,(*n* - 2)/2, *n*/2}. Therefore, each column of L_1 contains elements belonging to the set {-(*n* - 1), -(*n* - 3), … , -(*n* - 2i + 1), … , (*n* – 3), (*n* – 1), - *n*, -(*n* - 2), … , *n*} = {-*n*, -(*n* – 1), … ,(*n* – 1), *n*}. Clearly, these 2*n* + 1 elements are equally spaced. To prove orthogonality, let any two columns of L_1^* be x_{i1} and x_{i2}, *i* = 1, 2. Corresponding two columns in L_1 are $x_1 = \begin{pmatrix} x_{11} \\ x_{21} \end{pmatrix}$ and $x_2 = \begin{pmatrix} x_{12} \\ x_{22} \end{pmatrix}$. Now $x_1^T x_2 = x_{11}^T x_{12} + x_{21}^T x_{22} = 0$ since $x_{i1}^T x_{i2} = 0$ as $L_1^*, i = 1,2$ are orthogonal. Thus, L_1 is an OLH. Further, $L_1^* \subset L_1$ and $L_2^* \subset L_2$. Thus, L_1 is an NOL((2*n* + 1, *n*), m) with two layers L_1, L_1^* and an NOL((2*n* + 1, *n* + 1), *m*) with two layers L_1, L_2^*.

Example 1: *Let n=7, m=3.* Then using Theorem 1, we get a NOLH design as given in Table1

Table 1: *An NOL((15,7),3) and an NOL((15,8),3)*

Run#	x_1	x_2	x_3	*Run#*	x_1	x_2	x_3	*Run#*	x_1	x_2	x_3
1	*-6*	*6*	*4*	*6*	*4*	*2*	*-4*	*11*	*-1*	*-3*	*5*
2	*-4*	*0*	*-6*	*7*	*6*	*4*	*0*	*12*	*7*	*5*	*3*
3	*-2*	*-4*	*-2*	*8*	*-7*	*-5*	*-3*	*13*	*5*	*-7*	*1*
4	*0*	*-6*	*2*	*9*	*-5*	*7*	*-1*	*14*	*3*	*-1*	*-7*
5	*2*	*-2*	*6*	*10*	*-3*	*1*	*7*	*15*	*1*	*3*	*-5*

First 7 runs of the design given in Table 1 form the second layer of NOL((15,7),3) and last 8 runs form the second layer of NOL((15,8),3) and all the 15 runs form the first layer.

Theorem 1 permits construction of many smaller nested OLHs with two layers than the method of yang *et. al*. (2014). For example, *NOL*((15,7),3) and *NOL*((15,8),3) are perhaps the smallest nested OLH design for $m = 3$ in terms of run size. Method of yang *et. al*. (2014) requires at least 17 runs to get a nested OLH for $m = 3$. Again, to obtain a two-layer nested OLH for 5 and 6 factors using method of Yang *et al*. (2014) requires at least 33 runs whereas Thoerem 1 can obtain a nested OLH in 23 runs.

4.3 Construction of nested OLHs with three or more layers

In this section, we propose a method of construction of nested OLHs with $p > 2$ layers. As earlier, the construction depends on the existence of p OLHs with specific number of runs.

Theorem 2: Let L_1^* be an OLH (n, m), L_2^* be an OLH $(n + 1, m)$ and L_i^* be an OLH $(2^{(i-2)}n, m)$, $i = 3, 4,\ldots, p$ such that $n \equiv 0 \pmod 4$. Then $L_1 = (2^{p-1} L_1^{*T}, 2^{p-1} L_2^{*T}, 2^{p-2} L_3^{*T}, \ldots, 2^{p-i+1} L_i^{*T}, \ldots, 2^2 L_{p-1}^{*T}, 2 L_p^{*T})^T$ is an $NOL((n_1,n_2,\ldots,n_p),m)$ with p layers L_1, $L_2 = (2^{p-1} L_1^{*T}, 2^{p-1} L_2^{*T}, 2^{p-2} L_3^{*T}, \ldots, 2^{p-i+1} L_i^{*T}, \ldots, 2^2 L_{p-1}^{*T})^T, \ldots, L_i = (2^{p-1} L_1^{*T}, 2^{p-1} L_2^{*T}, 2^{p-2} L_3^{*T}, \ldots, 2^i L_{p-i+1}^{*T})^T, \ldots, L_{p-1} = (2^{p-1} L_1^{*T}, 2^{p-1} L_2^{*T})$, $L_p = 2^{p-1} L_1^*$ or $L_p = 2^{p-1} L_2^*$ and $n_1 = 2^{p-1} n+1$, $n_2 = 2^{p-2} n+1, \ldots, n_i = 2^{p-i} n + 1, \ldots, n_{p-1} = 2n + 1$, $n_p = n$ or $n + 1$, respectively.

Proof. First we prove one dimensional uniformity of the constructed design for each layer. One dimensional uniformity of p^{th} layer is obvious. Consider $(p - 1)^{th}$ layer $L_{p-1} = (2^{p-1} L_1^{*T}, 2^{p-1} L_2^{*T})$. Since $n \equiv 0$ (*mod* 4), let $n = 2^2 t$ where t is a positive integer. Then, each column of L_{p-1} contains elements belonging to set $\{-(2^p t - 2^{p-2}), -(2^p t - 3 \times 2^{p-2}), \ldots, (2^p t - 2^{p-2}), (2^p t - 3 \times 2^{p-2}), -2^p t, -(2^p t - 2 \times 2^{p-2}), \ldots, (2^p t - 2 \times 2^{p-2})\} = \{-2^p t, -(2^p t - 2^{p-2}), -(2^p t - 2 \times 2^{p-2}), \ldots, (2^p t - 2 \times 2^{p-2}), (2^p t - 2^{p-2}), 2^p t\}$. Clearly, the elements are equally spaced with spacing 2^{p-2}. Consider $(p - 2)(p-2)$th layer L_{p-2}. It may be verified easily that elements of each column of L_{p-2} are in the set $\{-2^p t, -(2^p t - 2^{p-3}), -(2^p t - 2 \times 2^{p-3}), \ldots, (2^p t - 2 \times 2^{p-3}), (2^p t - 2^{p-3}), 2^p t\}$, the spacing between the elements being 2^{p-3}. On similar lines, elements of each column of ith layer L_i, $i = 1, 2,\ldots, p - 1$ are in the set $\{-2^p t, -(2^p t - 2^{i-1}), -(2^p t - 2 \times 2^{i-1}), \ldots, (2^p t - 2 \times 2^{i-1}), (2^p t - 2^{i-1}), 2^p t\}$ with spacings 2^{i-1} between the elements. Proof of orthgonality is straightforward extension of the proof of orthogonality given in Theorem 1. Also $L_p \subset L_{p-1} \subset \ldots \subset L_2 \subset L_1$. This completes the proof.

Example 2: Let us construct a three layer nested OLH. Let $n = 12$, $m = 5$. Then an OLH(12,5), an OLH(13,5) and an OLH(24,5) are

$$L_1^* = \begin{pmatrix} -5.5 & -5.5 & -1.5 & -5.5 & -3.5 \\ -4.5 & -2.5 & -2.5 & 5.5 & 4.5 \\ -3.5 & 4.5 & 5.5 & -4.5 & -.5 \\ -2.5 & 0.5 & 0.5 & 0.5 & 0.5 \\ -1.5 & 2.5 & -.5 & 1.5 & 5.5 \\ -0.5 & 5.5 & 2.5 & 3.5 & -2.5 \\ 0.5 & 1.5 & -5.5 & 2.5 & -5.5 \\ 1.5 & -1.5 & 1.5 & -1.5 & 1.5 \\ 2.5 & -4.5 & 3.5 & 4.5 & -4.5 \\ 3.5 & -0.5 & -4.5 & -3.5 & 3.5 \\ 4.5 & 3.5 & -3.5 & -2.5 & -1.5 \\ 5.5 & -3.5 & 4.5 & -0.5 & 2.5 \end{pmatrix} \qquad L_2^* = \begin{pmatrix} -6 & -6 & -6 & 0 & -5 \\ -5 & 1 & 4 & -1 & 6 \\ -4 & 6 & -4 & 5 & 5 \\ -3 & 2 & 6 & -4 & -6 \\ -2 & -2 & 2 & 2 & -2 \\ -1 & 3 & 1 & 1 & -3 \\ 0 & 4 & -2 & -6 & 1 \\ 1 & -4 & -5 & -2 & 3 \\ 2 & -5 & 5 & 6 & 2 \\ 3 & -3 & 3 & -5 & 4 \\ 4 & -1 & 0 & 4 & -1 \\ 5 & 5 & -1 & 3 & -4 \\ 6 & 0 & -3 & -3 & 0 \end{pmatrix}$$

and

$$L_3^* = \begin{pmatrix} 7.5 & -2.5 & 9.5 & 11.5 & -10.5 \\ 9.5 & 7.5 & -2.5 & -10.5 & 8.5 \\ -2.5 & 9.5 & 7.5 & 8.5 & 11.5 \\ -11.5 & 10.5 & -8.5 & 7.5 & -2.5 \\ 10.5 & -8.5 & -11.50 & 9.5 & 7.5 \\ -8.5 & -11.5 & 10.5 & -2.5 & 9.5 \\ 3.5 & -1.5 & -0.5 & 5.5 & 6.5 \\ -1.5 & -0.5 & 3.5 & 6.5 & 4.5 \\ -0.5 & 3.5 & -1.5 & 4.5 & 5.5 \\ -6.5 & -5.5 & -4.5 & -1.5 & 3.5 \\ -5.5 & -4.5 & -6.5 & 3.5 & -0.5 \\ -4.5 & -6.5 & -5.5 & -0.5 & -1.5 \\ -7.5 & 2.5 & -9.5 & -11.5 & 10.5 \\ -9.5 & -7.5 & 2.5 & 10.5 & -8.5 \\ 2.5 & -9.5 & -7.5 & -8.5 & -11.5 \\ 11.5 & -10.5 & 8.5 & -7.5 & 2.5 \\ -10.5 & 8.5 & 11.5 & -9.5 & -7.5 \\ 8.5 & 11.5 & -10.5 & 2.5 & -9.5 \\ -3.5 & 1.5 & 0.5 & -5.5 & -6.5 \\ 1.5 & 0.5 & -3.5 & -6.5 & -4.5 \\ 0.5 & -3.5 & 1.5 & -4.5 & -5.5 \\ 6.5 & 5.5 & 4.5 & 1.5 & -3.5 \\ 5.5 & 4.5 & 6.5 & -3.5 & 0.5 \\ 4.5 & 6.5 & 5.5 & 0.5 & 1.5 \end{pmatrix}$$

, respectively.

Using Theorem 2, an NOL((49, 25, 12), 5) and an NOL((49, 25, 13), 5) are obtained and are given in Table 2.

Table 2: An *NOL*((49,25,12),5) and an *NOL*((49,25,13),5)

Run#	x_1	x_2	x_3	x_4	x_5	Run#	x_1	x_2	x_3	x_4	x_5
1	-22	-22	-6	-22	-14	25	24	0	-12	-12	0
2	-18	-10	-10	22	18	26	15	-5	19	23	-21
3	-14	18	22	-18	-2	27	19	15	-5	-21	17
4	-10	2	2	2	2	28	-5	19	15	17	23
5	-6	10	-2	6	22	29	-23	21	-17	15	-5
6	-2	22	10	14	-10	30	21	-17	-23	19	15
7	2	6	-22	10	-22	31	-17	-23	21	-5	19
8	6	-6	6	-6	6	32	7	-3	-1	11	13
9	10	-18	14	18	-18	33	-3	-1	7	13	9
10	14	-2	-18	-14	14	34	-1	7	-3	9	11
11	18	14	-14	-10	-6	35	-13	-11	-9	-3	7
12	22	-14	18	-2	10	36	-11	-9	-13	7	-1
13	-24	-24	-24	0	-20	37	-9	-13	-11	-1	-3
14	-20	4	16	-4	24	38	-15	5	-19	-23	21
15	-16	24	-16	20	20	39	-19	-15	5	21	-17
16	-12	8	24	-16	-24	40	5	-19	-15	-17	-23
17	-8	-8	8	8	-8	41	23	-21	17	-15	5
18	-4	12	4	4	-12	42	-21	17	23	-19	-15
19	0	16	-8	-24	4	43	17	23	-21	5	-19
20	4	-16	-20	-8	12	44	-7	3	1	-11	-13
21	8	-20	20	24	8	45	3	1	-7	-13	-9
22	12	-12	12	-20	16	46	1	-7	3	-9	-11

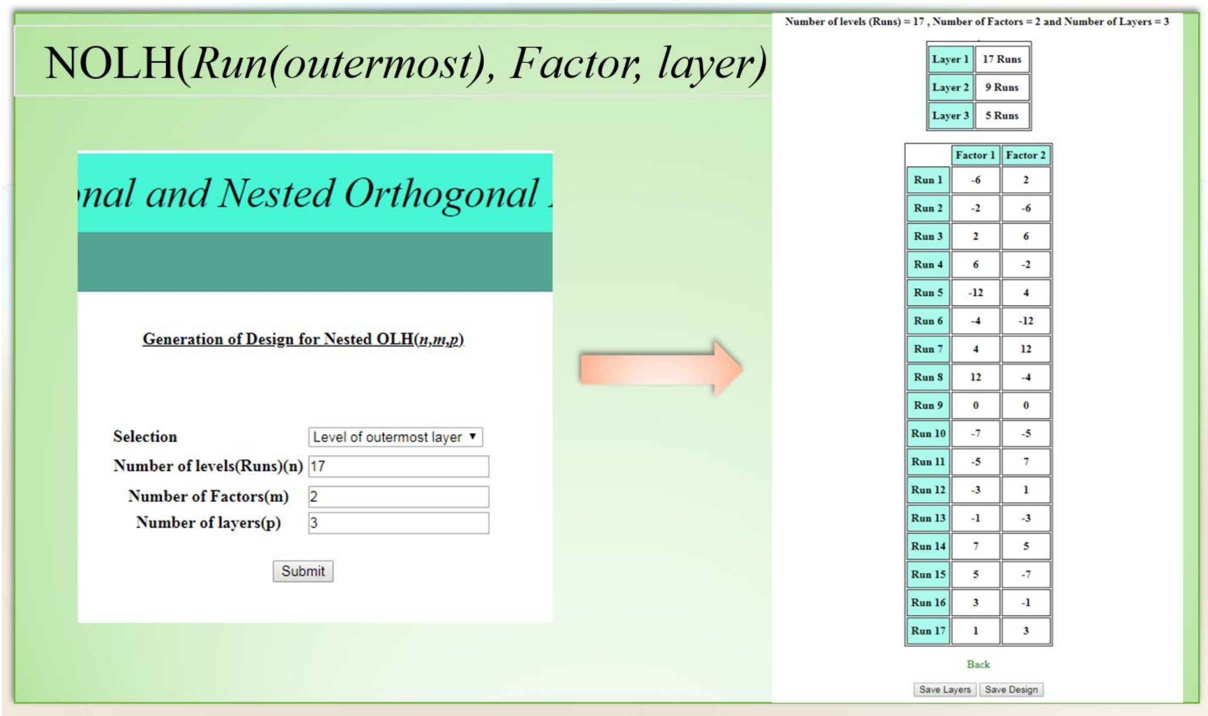
23	16	-4	0	16	-4	47	13	11	9	3	-7
24	20	20	-4	12	-16	48	11	9	13	-7	1
						49	9	13	11	1	3

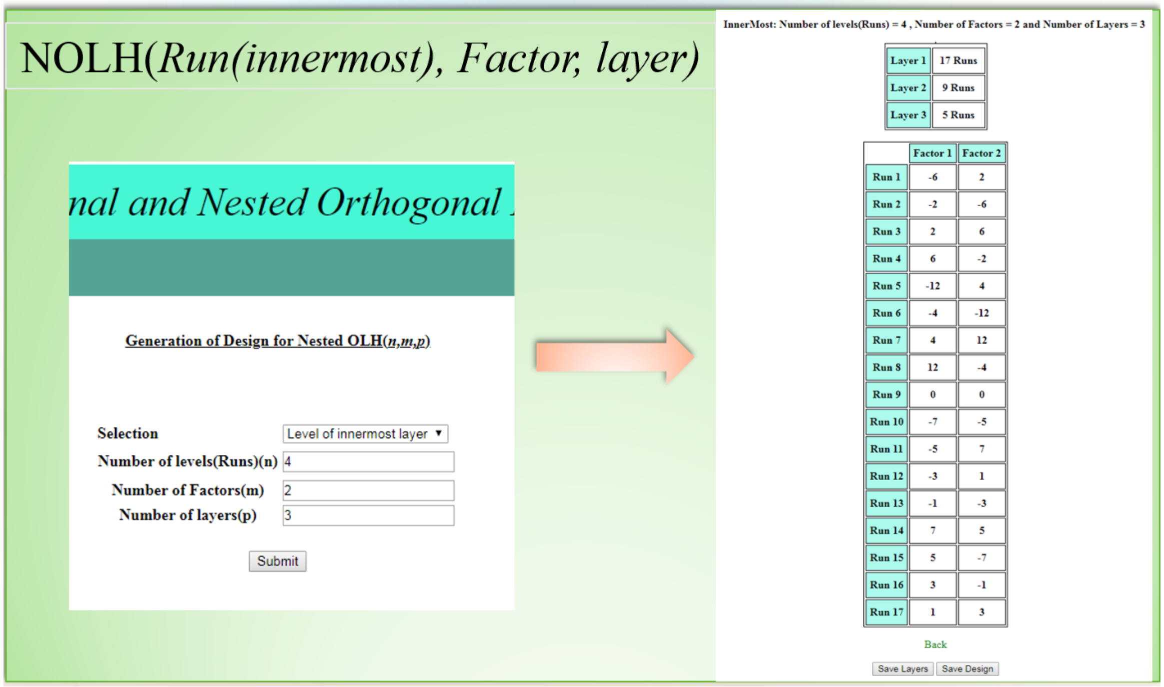
All 49 runs in Table 2 forms the first layer, first 25 runs form the second layer of both the NOLs and first 12 runs form the third layer of NOL((49, 25, 12), 5) and 13th to 25th runs form the third layer of NOL((49, 25, 13), 5).

Theorem 2 often can construct nested OLH with smaller number of runs than the available methods. A comparison of methods given by Yang *et al*.(2014) and proposed methods in this article for construction of nested OLHs with $p \geq 2$ layers in terms of smallest number of runs in the first layer is given in Table 3. Clearly, proposed methods permit more economical nested OLHs in terms of run size of the first layer for many values of m. When $m = 2^r$, $r = 2, 3, \ldots$, proposed methods give nested OLH designs with same minimum number of runs as given by the methods of Yang *et. al*.(2014).

4.4 Web application

In this section, we describe a web application developed for online generation of Nested Orthogonal Latin hypercube Designs with p layers nesting. The application has been developed using JSP language and STS (Java) platform. Some screen shots are given below:





4.5 Discussion

In this Chapter, we have proposed two general methods of constructing nested orthogonal Latin hypercube designs. The proposed methods have several advantages. Firstly, they permit construction of nested OLH for any value of m whenever constituent OLHs exist for that m. Secondly, existing OLHs can be effectively used to construct nested OLH. Thirdly, often it is possible to construct a nested OLH with smaller number of runs using the proposed methods as compared to the method of Yang et al. (2014). Moreover, nested OLHs constructed using proposed method are of second order whenever the constituent OLHs are of second order. For a definition of second order OLH, see Dey and Sarkar (2014); Parui *et al*., (2016).

Table 3: Minimum number of runs of nested OLHs using proposed methods (PM) and the methods of Yang et al. (2014). (YM)

p	m	YM	PM	p	m	YM	PM
2	2	9	9	5	2	65	65
	3	17	15		3	129	129
	4	17	17		4	129	129
	5	33	23		5	257	193
	6	33	23		6	257	193
3	2	17	17	6	2	129	129
	3	33	33		3	257	257
	4	33	33		4	257	257
	5	65	49		5	513	385
	6	65	49		6	513	385
4	2	33	33	p>2	2	$2^{p+1} + 1$	$2^{p+1} + 1$
	3	65	65		3	$2^{p+2} + 1$	$2^{p+2} + 1$
	4	65	65		4	$2^{p+2} + 1$	$2^{p+2} + 1$
	5	129	97		5	$4\times2^{p+1} + 1$	$3\times2^{p+1} + 1$
	6	129	97		6	$4\times2^{p+1} + 1$	$3\times2^{p+1} + 1$

SUMMARY

Latin hypercube designs (LHDs) are commonly used in designing computer experiments. In recent years, several methods of constructing orthogonal Latin hypercube designs have been proposed in the literature. In this article, the methods of construction of orthogonal Latin hypercube designs for six factors for any permissible number of runs have been obtained. Further, two new series of second order orthogonal Latin hypercube designs for six factors have been given. A Catalogue of Orthogonal Latin hypercube Designs of 1st order and 2nd order with m (≤ 6) factors and n (≤ 20) runs is also presented in Appendix I and Appendix II respectively. The 1st order OLH(n, m) designs ensure independence of estimates of linear effects when a first order model is fitted whereas 2nd order OLH design ensures that not only the estimates of linear effects are mutually uncorrelated but they are also are uncorrelated with the estimates of quadratic and interaction effects in a second order model. A web application of generation of these designs has also been developed and presented in Chapter 2.

Latin hypercube designs are widely used space-filling designs. Choice of a good Latin hypercube closely depends upon two criteria viz. space-filling criteria and orthogonality. Space-filling criterion provides maximum coverage to the whole design space and orthogonality criterion helps to estimate parameters independently.In the Chapter III, construction methods of Orthogonal and nearly orthogonal space filling Latin Hypercube Designs has been described. A Catalogue of Orthogonal and nearly orthogonal space filling Latin Hypercube Designs with m (≤ 6) factors and n (≤ 20) runs is presented in Appendix III(a). The orthogonality value and space filling values of the designs presented in Appendix III(a) is presented in Appendix III(b). To determine a Latin hypercube design with respect to space-filling property, three criterion like entropy criterion, ϕ_p criterion and central L_2 discrepancy criterion was used. A web application of generation of these designs has also been developed and presented in Chapter 3.

Sometimes computer experiments may be performed with two or more different accuracy levels to study a complex physical system. A Latin hypercube design L_1 is said to be nested with $p \geq 2$ layers when it contains a smaller Latin hypercube design L_2 as a subset, L_2 contains a smaller Latin hypercube design L_3 as a subset and so on till L_{p-1} contains a smaller Latin hypercube design L_p as a subset. Clearly, $n_p < n_{p-1} < \ldots < n_1$. Further, if each L_i is an OLH, then L_1 is a nested OLH and is denoted as NOL((n_1, n_2,…,n_p), m). The p layers of the nested OLH are L_1, L_2,…,L_p,

respectively. Recently, Yang *et al*. (2014) proposed methods of construction of nested OLH. They used a special type of $m \times m$ orthogonal design (OD) to construct nested OLH. Though their methods are quite appealing, the methods have two limitations. Firstly, the methods work for $m = 2^r$, r being a positive integer and secondly NOLs obtained through their proposed methods often have larger number of runs for a given m. In this article, we propose two very simple methods of construction of nested OLHs with p layers. The methods use existing OLHs to obtain a nested OLH. Using the proposed methods, we obtain many new nested OLHs with p layers. In the Chapter IV, construction methods of Nested Orthogonal Latin Hypercube (NOLH) Designs has been described. Two general methods of constructing nested orthogonal Latin hypercube designs have been developed. First method deals with 2 layers of NOLH and the second methods deals with three or more layers of NOLH. The methods give many new nested orthogonal Latin hypercube designs with fewer number of runs as compared to existing nested orthogonal Latin hypercube designs. A Catalogue of Nested Orthogonal Latinhypercube Designs of $p(\leq 4)$ layers with $m(\leq 6)$ factors and $n(< 100)$ runs is presented in Appendix IV. A web application of generation of these designs has also been developed and presented in Chapter 4.

सारांश

लैटिन हाइपरक्यूब अभिकल्पनाओं (LHDs) का उपयोग सामान्यतः संगणक परीक्षणों को अभिकल्पित करने में होता है। हाल के वर्षों में, सहित्य में लाम्बिक लैटिन हाइपरक्यूब (ओ एल एच) अभिकल्पनाएं संरचित करने की बहुत सी पद्धतियां प्रस्तावित की गई हैं। इस प्रस्तुति में, रनों की किसी भी अनुमत संख्या हेतु छः कारकों के लिए लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाओं की संरचना हेतु पद्धतियां प्राप्त की गई हैं। इसके अतिरिक्त, छः कारकों हेतु द्वितीय क्रम लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाअें की दो नई श्रंखलाएं भी दी गई हैं। m(<=6) कारक और n(<=20) रन वाले प्रथम एवं द्वितीय क्रम की लाम्बिक लैटिन हाइपरक्यूब की एक सूची भी क्रमषः परिषिष्ट I एवं परिषिष्ट II में दी गई है। प्रथम क्रम ओ एल एच (n,m) अभिकल्पनाएं रेखीय प्रभावों के आंकलन सुनिष्चित करती है जब प्रथम कम मॉडल लगाया जाता है जबकि द्वितीय क्रम ओ एल एच अभिकल्पनाएं न केवल एक–दूसरे से असहसंबंधित रेखीय प्रभावों के आंकलन सुनिष्चित करती है बल्कि एक द्वितीय क्रम मॉडल में द्विघात एवं अंतःक्रिया प्रभावों के आंकलन सहित वे असहसंबंधित हैं यह भी सुनिष्चित करती है। इन अभिकल्पनाओं की उत्पत्ति का एक वैब अनुप्रयोग भी विकसित किया गया है और अध्याय 2 में प्रस्तुत किया गया है।

लैटिन हाइपरक्यूब अभिकल्पनाओं का उपयोग वृहद रुप से स्थान–भरने वाली अभिकल्पनाओं की तरह किया जाता है। एक अच्छी लैटिन हाइपरक्यूब का विकल्प मुख्य रुप से दो मानदंडों जैसे कि स्थान–भरना मानदंड एवं लाम्बिकता पर निर्भर करता है। स्थान–भरना मानदंड सम्पूर्ण अभिकल्पना स्थान की अधिकतम कार्यक्षेत्र व्याप्ति उपलब्ध कराता है और लाम्बिकता मानदंड प्राचलों के स्वतंत्र आंकलन में मदद करता है। तृतीय अध्याय में, लाम्बिक एवं लगभग लाम्बिक स्थान–भरने वाली लैटिन हाइपरक्यूब अभिकल्पनाओं की संरचना पद्धति का वर्णन है। m(<=6) कारक और n(<=20) रन वाली लाम्बिक एवं लगभग लाम्बिक स्थान–भरने वाली लैटिन हाइपरक्यूब अभिकल्पनाओं की एक सूची परिषिष्ट III (a) में दी गई है। परिषिष्ट III(a) में दिये गए अभिकल्पनाओं के मान एवं स्थान–भरने वाले मान परिषिष्ट III(b) में दिये गए हैं। स्थान–भरने के गुण के सापेक्ष लैटिन हाइपरक्यूब अभिकल्पनाएं ज्ञात करने के लिए तीन मानदंडों जैसे कि उत्क्रम–माप मानदंड, ϕ_p मानदंड एवं केन्द्रीय L_2 विसंगति मानदंड का उपयोग किया गया था। इन अभिकल्पनाओं की उत्पत्ति का एक वैब अनुप्रयोग भी विकसित किया गया है और अध्याय 3 में प्रस्तुत किया गया है।

कभी–कभी जटिल भौतिक प्रणाली का अध्ययन करने के लिए दो या दो से अधिक परिशुद्धता स्तर वाले संगणक परीक्षण किए जाते हैं । कियान (2009) ने नीड़ित लैटिन हाइपरक्यूब अभिकल्पनाएं प्रस्तावित की जो इस प्रकार के परीक्षण संचालित करने में उपयोगी हैं। p>=2 परतों वाली किसी लैटिन हाइपरक्यूब अभिकल्पना L_1 को नीड़ित तब कहा जाता है जब इसमें L_2 लैटिन हाइपरक्यूब अभिकल्पना उपसमुच्चय जैसी हो, L_2 में एक छोटी लैटिन हाइपरक्यूब अभिकल्पना L_3 उपसमुच्चय जैसी हो और यह क्रम तब तक चलता रहता है जब तक L_{p-1} में एक छोटी लैटिन हाइपरक्यूब अभिकल्पना L_p उपसमुच्चय जैसी हो जाए। स्पष्टतः $n_p<n_{p-1}<\ldots<n_1$ । इसके आगे यदि प्रत्येक L_1 एक लाम्बिक लैटिन हाइपरक्यूब है तो L_1 एक नीड़ित लाम्बिक लैटिन हाइपरक्यूब है और यह $NOL((n_1,n_2,\ldots,n_p),m)$ द्वारा प्रदर्शित है।

नीड़ित लाम्बिक लैटिन हाइपरक्यूब की p परतें क्रमशः $L_1, L_2, ..., L_p$ हैं। हाल ही में, यांग इत्यादि (2014) ने नीड़ित लाम्बिक लैटिन हाइपरक्यूब की संरचना पद्धति प्रस्तावित की। उन्होंने नीड़ित लाम्बिक लैटिन हाइपरक्यूब की संरचना करने के लिए एक विशेष प्रकार की mxm लाम्बिक अभिकल्पना का उपयोग किया। यद्यपि उनकी पद्धतियाँ बहुत अच्छी हैं फिर भी उनकी दो सीमाएँ हैं। प्रथम तो यह कि यह पद्धति $m=2^r$ के लिए काम करती है जहाँ पर r एक धनात्मक पूर्णांक है और दूसरी यह कि उनके रनों प्रस्तावित पद्धति से प्राप्त NOLS दिए गए के लिए अक्सर रनों की संख्या बहुत अधिक होती है।

इस प्रस्तुति में, हमने p परतों वाली नीड़ित लाम्बिक लैटिन हाइपरक्यूब्स की संरचना की बहुत ही सरल दो पद्धतियाँ विकसित की हैं। पद्धतियाँ नीड़ित लाम्बिक लैटिन हाइपरक्यूब प्राप्त करने के लिए वर्तमान लाम्बिक लैटिन हाइपरक्यूब का उपयोग करती हैं। प्रस्तावित पद्धतियों का उपयोग करते हुए हम p परतों वाली बहुत सी नई नीड़ित लाम्बिक लैटिन हाइपरक्यूब प्राप्त करते हैं। अध्याय V में, नीड़ित लाम्बिक लैटिन

हाइपरक्यूब अभिकल्पनाओं की संरचना पद्धति का वर्णन किया गया है। नीड़ित लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाओं की संरचना करने की दो सामान्य पद्धतियाँ विकसित की गई हैं। पहली पद्धति का सरोकार दो परतों वाली नीड़ित लाम्बिक लैटिन हाइपरक्यूब व दूसरी पद्धति का सरोकार तीन अथवा उससे अधिक परतों वाली नीड़ित लाम्बिक लैटिन हाइपरक्यूब से है। नई पद्धति वर्तमान लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाओं की अपेक्षा कम रनों द्वारा बहुत सी नई नीड़ित लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाएं सृजित करती है। m(<=6) कारकों और n(<=100) रन सहित p(<=4) परतों वाली नीड़ित लाम्बिक लैटिन हाइपरक्यूब अभिकल्पनाओं की परिषिष्ट IV में दी गई है। इन अभिकल्पनाओं की उत्पत्ति का वैब अनुप्रयोग भी विकसित किया गया है और अध्याय 4 में दिया गया है।

Appendix I: Catalogue of Orthogonal Latin hypercube Designs of 1st order with m (≤ 6) factors and n (≤ 20) runs

Number of factors =2

	Factor 1	Factor 2
Run 1	-1	0.5
Run 2	-0.5	-1
Run 3	0.5	1
Run 4	1	-0.5

	Factor 1	Factor 2
Run 1	-2	1
Run 2	-1	-2
Run 3	1	2
Run 4	2	-1
Run 5	0	0

	Factor 1	Factor 2
Run 1	-3	3
Run 2	-2	0
Run 3	-1	-2
Run 4	0	-3
Run 5	1	-1
Run 6	2	1
Run 7	3	2

	Factor 1	Factor 2
Run 1	-3.5	-2.5
Run 2	-2.5	3.5
Run 3	-1.5	0.5
Run 4	-0.5	-1.5
Run 5	3.5	2.5
Run 6	2.5	-3.5
Run 7	1.5	-0.5
Run 8	0.5	1.5

	Factor 1	Factor 2
Run 1	-4	-2
Run 2	-3	4
Run 3	-2	-3
Run 4	-1	3
Run 5	0	-4
Run 6	1	2
Run 7	2	0
Run 8	3	1
Run 9	4	-1

	Factor 1	Factor 2
Run 1	-5	-4
Run 2	-4	2
Run 3	-3	-2
Run 4	-2	3
Run 5	-1	4
Run 6	0	-5
Run 7	1	5
Run 8	2	-1
Run 9	3	0
Run 10	4	1
Run 11	5	-3

	Factor 1	Factor 2
Run 1	-5.5	-5.5
Run 2	-4.5	-2.5
Run 3	-3.5	4.5
Run 4	-2.5	0.5
Run 5	-1.5	2.5
Run 6	-0.5	5.5
Run 7	0.5	1.5
Run 8	1.5	-1.5
Run 9	2.5	-4.5
Run 10	3.5	-0.5
Run 11	4.5	3.5
Run 12	5.5	-3.5

	Factor 1	Factor 2
Run 1	-6	-6
Run 2	-5	1
Run 3	-4	6
Run 4	-3	2
Run 5	-2	-2
Run 6	-1	3
Run 7	0	4
Run 8	1	-4
Run 9	2	-5
Run 10	3	-3
Run 11	4	-1
Run 12	5	5
Run 13	6	0

	Factor 1	Factor 2
Run 1	-7	4
Run 2	-6	3
Run 3	-5	-6

	Factor 1	Factor 2
Run 1	-7.5	6.5
Run 2	-6.5	-7.5
Run 3	-5.5	4.5

	Factor 1	Factor 2
Run 1	-8	7
Run 2	-7	-8
Run 3	-6	5

	Factor 1	Factor 2
Run 1	-9	-9
Run 2	-8	5
Run 3	-7	-6

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

Run 4	-4.5	-5.5
Run 5	-3.5	2.5
Run 6	-2.5	-3.5
Run 7	-1.5	0.5
Run 8	-0.5	-1.5
Run 9	7.5	-6.5
Run 10	6.5	7.5
Run 11	5.5	-4.5
Run 12	4.5	5.5
Run 13	3.5	-2.5
Run 14	2.5	3.5
Run 15	1.5	-0.5
Run 16	0.5	1.5

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

Run 4	-6	-3
Run 5	-5	9
Run 6	-4	7
Run 7	-3	-2
Run 8	-2	6
Run 9	-1	-4
Run 10	0	4
Run 11	1	0
Run 12	2	8
Run 13	3	-8
Run 14	4	-7
Run 15	5	-5
Run 16	6	1
Run 17	7	2
Run 18	8	3
Run 19	9	-1

	Factor 1	**Factor 2**
Run 1	-9.5	6.5
Run 2	-8.5	-6.5
Run 3	-7.5	-4.5
Run 4	-6.5	4.5
Run 5	-5.5	-7.5
Run 6	-4.5	9.5
Run 7	-3.5	7.5
Run 8	-2.5	-1.5
Run 9	-1.5	-9.5
Run 10	-0.5	0.5
Run 11	0.5	-0.5
Run 12	1.5	-8.5
Run 13	2.5	2.5
Run 14	3.5	5.5
Run 15	4.5	1.5
Run 16	5.5	8.5
Run 17	6.5	-2.5
Run 18	7.5	-5.5
Run 19	8.5	-3.5
Run 20	9.5	3.5

Number of factors =3

	Factor1	Factor2	Factor3
Run1	-3	3	2
Run2	-2	0	-3
Run3	-1	-2	-1
Run4	0	-3	1
Run5	1	-1	3
Run6	2	1	-2
Run7	3	2	0

	Factor1	Factor2	Factor3
Run1	-3.5	-2.5	-1.5
Run2	-2.5	3.5	-0.5
Run3	-1.5	0.5	3.5
Run4	-0.5	-1.5	2.5
Run5	3.5	2.5	1.5
Run6	2.5	-3.5	0.5
Run7	1.5	-0.5	-3.5
Run8	0.5	1.5	-2.5

	Factor1	Factor2	Factor3
Run1	-4	-2	0
Run2	-3	4	2
Run3	-2	-3	-4
Run4	-1	3	-2
Run5	0	-4	4
Run6	1	2	-1
Run7	2	0	3
Run8	3	1	1
Run9	4	-1	-3

	Factor1	Factor2	Factor3
Run1	-5	-4	-5
Run2	-4	2	-1
Run3	-3	-2	4
Run4	-2	3	-3
Run5	-1	4	2
Run6	0	-5	5
Run7	1	5	3
Run8	2	-1	1
Run9	3	0	0
Run10	4	1	-4
Run11	5	-3	-2

	Factor1	Factor2	Factor3
Run1	-5.5	-5.5	-1.5
Run2	-4.5	-2.5	-2.5
Run3	-3.5	4.5	5.5
Run4	-2.5	0.5	0.5
Run5	-1.5	2.5	-0.5
Run6	-0.5	5.5	2.5
Run7	0.5	1.5	-5.5
Run8	1.5	-1.5	1.5
Run9	2.5	-4.5	3.5
Run10	3.5	-0.5	-4.5
Run11	4.5	3.5	-3.5
Run 12	5.5	-3.5	4.5

	Factor1	Factor2	Factor3
Run 1	-6	-6	-6
Run 2	-5	1	4
Run 3	-4	6	-4
Run 4	-3	2	6
Run 5	-2	-2	2
Run 6	-1	3	1
Run 7	0	4	-2
Run 8	1	-4	-5
Run 9	2	-5	5
Run 10	3	-3	3
Run 11	4	-1	0
Run 12	5	5	-1
Run 13	6	0	-3

	Factor1	Factor2	Factor3
Run 1	-7	4	-7
Run 2	-6	3	5
Run 3	-5	-6	-2
Run 4	-4	1	4
Run 5	-3	0	-4
Run 6	-2	-2	3
Run 7	-1	-7	6

	Factor1	Factor2	Factor3
Run 1	-7.5	6.5	4.5
Run 2	-6.5	-7.5	5.5
Run 3	-5.5	4.5	-6.5
Run 4	-4.5	-5.5	-7.5
Run 5	-3.5	2.5	0.5
Run 6	-2.5	-3.5	1.5
Run 7	-1.5	0.5	-2.5

	Factor1	Factor2	Factor3
Run 1	-8	7	5
Run 2	-7	-8	6
Run 3	-6	5	-7
Run 4	-5	-6	-8
Run 5	-4	3	1
Run 6	-3	-4	2
Run 7	-2	1	-3

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

Run 8	-0.5	-1.5	-3.5
Run 9	7.5	-6.5	-4.5
Run 10	6.5	7.5	-5.5
Run 11	5.5	-4.5	6.5
Run 12	4.5	5.5	7.5
Run 13	3.5	-2.5	-0.5
Run 14	2.5	3.5	-1.5
Run 15	1.5	-0.5	2.5
Run 16	0.5	1.5	3.5

Run 8	-1	-2	-4
Run 9	8	-7	-5
Run 10	7	8	-6
Run 11	6	-5	7
Run 12	5	6	8
Run 13	4	-3	-1
Run 14	3	4	-2
Run 15	2	-1	3
Run 16	1	2	4
Run17	0	0	0

	Factor1	**Factor2**	**Factor3**
Run 1	-9	-9	-9
Run 2	-8	5	5
Run 3	-7	-6	-5
Run 4	-6	-3	0
Run 5	-5	9	6
Run 6	-4	7	9
Run 7	-3	-2	3
Run 8	-2	6	-8
Run 9	-1	-4	4
Run 10	0	4	-6
Run 11	1	0	-2
Run 12	2	8	-3
Run 13	3	-8	8
Run 14	4	-7	2
Run 15	5	-5	7
Run 16	6	1	-4
Run 17	7	2	-7
Run 18	8	3	-1
Run 19	9	-1	1

	Factor1	**Factor2**	**Factor3**
Run 1	-9.5	6.5	-9.5
Run 2	-8.5	-6.5	4.5
Run 3	-7.5	-4.5	5.5
Run 4	-6.5	4.5	-1.5
Run 5	-5.5	-7.5	0.5
Run 6	-4.5	9.5	-0.5
Run 7	-3.5	7.5	-4.5
Run 8	-2.5	-1.5	3.5
Run 9	-1.5	-9.5	-3.5
Run 10	-0.5	0.5	7.5
Run 11	0.5	-0.5	8.5
Run 12	1.5	-8.5	-5.5
Run 13	2.5	2.5	-2.5
Run 14	3.5	5.5	1.5
Run 15	4.5	1.5	-7.5
Run 16	5.5	8.5	9.5
Run 17	6.5	-2.5	-6.5
Run 18	7.5	-5.5	6.5
Run 19	8.5	-3.5	-8.5
Run20	9.5	3.5	2.5

Number of factors =4

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-3.5	-2.5	-1.5	-0.5
Run 2	-2.5	3.5	-0.5	1.5
Run 3	-1.5	0.5	3.5	-2.5
Run 4	-0.5	-1.5	2.5	3.5
Run 5	3.5	2.5	1.5	0.5
Run 6	2.5	-3.5	0.5	-1.5
Run 7	1.5	-0.5	-3.5	2.5
Run 8	0.5	1.5	-2.5	-3.5

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-4	-2	0	-3
Run 2	-3	4	2	1
Run 3	-2	-3	-4	-1
Run 4	-1	3	-2	3
Run 5	0	-4	4	4
Run 6	1	2	-1	0
Run 7	2	0	3	-2
Run 8	3	1	1	-4
Run 9	4	-1	-3	2

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-5	-4	-5	-5
Run 2	-4	2	-1	3
Run 3	-3	-2	4	5
Run 4	-2	3	-3	4
Run 5	-1	4	2	-4
Run 6	0	-5	5	-2
Run 7	1	5	3	-3
Run 8	2	-1	1	1
Run 9	3	0	0	-1
Run 10	4	1	-4	0
Run 11	5	-3	-2	2

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-5.5	-5.5	-1.5	-5.5
Run 2	-4.5	-2.5	-2.5	5.5
Run 3	-3.5	4.5	5.5	-4.5
Run 4	-2.5	0.5	0.5	0.5
Run 5	-1.5	2.5	-0.5	1.5
Run 6	-0.5	5.5	2.5	3.5
Run 7	0.5	1.5	-5.5	2.5
Run 8	1.5	-1.5	1.5	-1.5
Run 9	2.5	-4.5	3.5	4.5
Run 10	3.5	-0.5	-4.5	-3.5
Run 11	4.5	3.5	-3.5	-2.5
Run 12	5.5	-3.5	4.5	-0.5

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-6	-6	-6	0
Run 2	-5	1	4	-1
Run 3	-4	6	-4	5
Run 4	-3	2	6	-4
Run 5	-2	-2	2	2
Run 6	-1	3	1	1
Run 7	0	4	-2	-6
Run 8	1	-4	-5	-2
Run 9	2	-5	5	6
Run 10	3	-3	3	-5
Run 11	4	-1	0	4
Run 12	5	5	-1	3
Run 13	6	0	-3	-3

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

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-7.5	6.5	4.5	5.5
Run 2	-6.5	-7.5	5.5	-4.5
Run 3	-5.5	4.5	-6.5	-7.5
Run 4	-4.5	-5.5	-7.5	6.5
Run 5	-3.5	2.5	0.5	-1.5
Run 6	-2.5	-3.5	1.5	0.5
Run 7	-1.5	0.5	-2.5	3.5
Run 8	-0.5	-1.5	-3.5	-2.5
Run 9	7.5	-6.5	-4.5	-5.5
Run 10	6.5	7.5	-5.5	4.5
Run 11	5.5	-4.5	6.5	7.5
Run 12	4.5	5.5	7.5	-6.5
Run 13	3.5	-2.5	-0.5	1.5
Run 14	2.5	3.5	-1.5	-0.5
Run 15	1.5	-0.5	2.5	-3.5
Run 16	0.5	1.5	3.5	2.5

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

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-9	-9	-9	-8
Run 2	-8	5	5	4
Run 3	-7	-6	-5	5
Run 4	-6	-3	0	-9
Run 5	-5	9	6	1
Run 6	-4	7	9	-5
Run 7	-3	-2	3	3
Run 8	-2	6	-8	6
Run 9	-1	-4	4	9
Run 10	0	4	-6	-2
Run 11	1	0	-2	-4
Run 12	2	8	-3	-1
Run 13	3	-8	8	0
Run 14	4	-7	2	8
Run 15	5	-5	7	-3
Run 16	6	1	-4	7
Run 17	7	2	-7	2
Run 18	8	3	-1	-6
Run 19	9	-1	1	-7

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-9.5	6.5	-9.5	-8.5
Run 2	-8.5	-6.5	4.5	8.5
Run 3	-7.5	-4.5	5.5	2.5
Run 4	-6.5	4.5	-1.5	-4.5
Run 5	-5.5	-7.5	0.5	-7.5
Run 6	-4.5	9.5	-0.5	5.5
Run 7	-3.5	7.5	-4.5	4.5
Run 8	-2.5	-1.5	3.5	-1.5
Run 9	-1.5	-9.5	-3.5	3.5
Run 10	-0.5	0.5	7.5	-5.5
Run 11	0.5	-0.5	8.5	-0.5
Run 12	1.5	-8.5	-5.5	6.5
Run 13	2.5	2.5	-2.5	-6.5
Run 14	3.5	5.5	1.5	1.5
Run 15	4.5	1.5	-7.5	0.5
Run 16	5.5	8.5	9.5	7.5
Run 17	6.5	-2.5	-6.5	9.5
Run 18	7.5	-5.5	6.5	-9.5
Run 19	8.5	-3.5	-8.5	-3.5
Run 20	9.5	3.5	2.5	-2.5

Number of factors =5

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-4	-2	0	-3	3
Run 2	-3	4	2	1	-2
Run 3	-2	-3	-4	-1	-3
Run 4	-1	3	-2	3	4
Run 5	0	-4	4	4	0
Run 6	1	2	-1	0	-4
Run 7	2	0	3	-2	-1
Run 8	3	1	1	-4	2
Run 9	4	-1	-3	2	1

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-5	-4	-5	-5	-3
Run 2	-4	2	-1	3	4
Run 3	-3	-2	4	5	-4
Run 4	-2	3	-3	4	1
Run 5	-1	4	2	-4	3
Run 6	0	-5	5	-2	5
Run 7	1	5	3	-3	-5
Run 8	2	-1	1	1	-2
Run 9	3	0	0	-1	0
Run 10	4	1	-4	0	2
Run 11	5	-3	-2	2	-1

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-5.5	-5.5	-1.5	-5.5	-3.5
Run 2	-4.5	-2.5	-2.5	5.5	4.5
Run 3	-3.5	4.5	5.5	-4.5	-0.5
Run 4	-2.5	0.5	0.5	0.5	0.5
Run 5	-1.5	2.5	-0.5	1.5	5.5
Run 6	-0.5	5.5	2.5	3.5	-2.5
Run 7	0.5	1.5	-5.5	2.5	-5.5
Run 8	1.5	-1.5	1.5	-1.5	1.5
Run 9	2.5	-4.5	3.5	4.5	-4.5
Run 10	3.5	-0.5	-4.5	-3.5	3.5
Run 11	4.5	3.5	-3.5	-2.5	-1.5
Run 12	5.5	-3.5	4.5	-0.5	2.5

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-6	-6	-6	0	-5
Run 2	-5	1	4	-1	6
Run 3	-4	6	-4	5	5
Run 4	-3	2	6	-4	-6
Run 5	-2	-2	2	2	-2
Run 6	-1	3	1	1	-3
Run 7	0	4	-2	-6	1
Run 8	1	-4	-5	-2	3
Run 9	2	-5	5	6	2
Run 10	3	-3	3	-5	4
Run 11	4	-1	0	4	-1
Run 12	5	5	-1	3	-4
Run 13	6	0	-3	-3	0

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

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-7.5	6.5	4.5	5.5	0.5
Run 2	-6.5	-7.5	5.5	-4.5	1.5
Run 3	-5.5	4.5	-6.5	-7.5	2.5
Run 4	-4.5	-5.5	-7.5	6.5	3.5
Run 5	-3.5	2.5	0.5	-1.5	-4.5
Run 6	-2.5	-3.5	1.5	0.5	-5.5
Run 7	-1.5	0.5	-2.5	3.5	-6.5
Run 8	-0.5	-1.5	-3.5	-2.5	-7.5
Run 9	7.5	-6.5	-4.5	-5.5	-0.5
Run 10	6.5	7.5	-5.5	4.5	-1.5
Run 11	5.5	-4.5	6.5	7.5	-2.5
Run 12	4.5	5.5	7.5	-6.5	-3.5
Run 13	3.5	-2.5	-0.5	1.5	4.5
Run 14	2.5	3.5	-1.5	-0.5	5.5
Run 15	1.5	-0.5	2.5	-3.5	6.5
Run 16	0.5	1.5	3.5	2.5	7.5

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

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-9	-9	-9	-8	2
Run 2	-8	5	5	4	-8
Run 3	-7	-6	-5	5	0
Run 4	-6	-3	0	-9	-4
Run 5	-5	9	6	1	7
Run 6	-4	7	9	-5	-2
Run 7	-3	-2	3	3	-9
Run 8	-2	6	-8	6	6
Run 9	-1	-4	4	9	5
Run 10	0	4	-6	-2	9
Run 11	1	0	-2	-4	-5
Run 12	2	8	-3	-1	3
Run 13	3	-8	8	0	4
Run 14	4	-7	2	8	-1
Run 15	5	-5	7	-3	8
Run 16	6	1	-4	7	-7
Run 17	7	2	-7	2	-3
Run 18	8	3	-1	-6	-6
Run 19	9	-1	1	-7	1

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-9.5	6.5	-9.5	-8.5	-9.5
Run 2	-8.5	-6.5	4.5	8.5	-5.5
Run 3	-7.5	-4.5	5.5	2.5	5.5
Run 4	-6.5	4.5	-1.5	-4.5	7.5
Run 5	-5.5	-7.5	0.5	-7.5	-6.5
Run 6	-4.5	9.5	-0.5	5.5	6.5
Run 7	-3.5	7.5	-4.5	4.5	-3.5
Run 8	-2.5	-1.5	3.5	-1.5	0.5
Run 9	-1.5	-9.5	-3.5	3.5	3.5
Run 10	-0.5	0.5	7.5	-5.5	-1.5
Run 11	0.5	-0.5	8.5	-0.5	-0.5
Run 12	1.5	-8.5	-5.5	6.5	4.5
Run 13	2.5	2.5	-2.5	-6.5	1.5
Run 14	3.5	5.5	1.5	1.5	9.5
Run 15	4.5	1.5	-7.5	0.5	8.5
Run 16	5.5	8.5	9.5	7.5	-7.5
Run 17	6.5	-2.5	-6.5	9.5	-8.5
Run 18	7.5	-5.5	6.5	-9.5	2.5
Run 19	8.5	-3.5	-8.5	-3.5	-2.5
Run 20	9.5	3.5	2.5	-2.5	-4.5

Number of factors =6

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-5	-4	-5	-5	-3	0
Run 2	-4	2	-1	3	4	5
Run 3	-3	-2	4	5	-4	-2
Run 4	-2	3	-3	4	1	-4
Run 5	-1	4	2	-4	3	2
Run 6	0	-5	5	-2	5	-3
Run 7	1	5	3	-3	-5	-1
Run 8	2	-1	1	1	-2	3
Run 9	3	0	0	-1	0	1
Run 10	4	1	-4	0	2	-5
Run 11	5	-3	-2	2	-1	4

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-5.5	-5.5	-1.5	-5.5	-3.5	-3.5
Run 2	-4.5	-2.5	-2.5	5.5	4.5	0.5
Run 3	-3.5	4.5	5.5	-4.5	-0.5	1.5
Run 4	-2.5	0.5	0.5	0.5	0.5	5.5
Run 5	-1.5	2.5	-0.5	1.5	5.5	-4.5
Run 6	-0.5	5.5	2.5	3.5	-2.5	-1.5
Run 7	0.5	1.5	-5.5	2.5	-5.5	-2.5
Run 8	1.5	-1.5	1.5	-1.5	1.5	2.5
Run 9	2.5	-4.5	3.5	4.5	-4.5	3.5
Run 10	3.5	-0.5	-4.5	-3.5	3.5	4.5
Run 11	4.5	3.5	-3.5	-2.5	-1.5	-0.5
Run 12	5.5	-3.5	4.5	-0.5	2.5	-5.5

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-6	-6	-6	0	-5	-1
Run 2	-5	1	4	-1	6	5
Run 3	-4	6	-4	5	5	-2
Run 4	-3	2	6	-4	-6	2
Run 5	-2	-2	2	2	-2	-4
Run 6	-1	3	1	1	-3	3
Run 7	0	4	-2	-6	1	-5
Run 8	1	-4	-5	-2	3	4
Run 9	2	-5	5	6	2	1
Run 10	3	-3	3	-5	4	-6
Run 11	4	-1	0	4	-1	-3
Run 12	5	5	-1	3	-4	0
Run 13	6	0	-3	-3	0	6

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

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-7.5	6.5	4.5	5.5	0.5	-3.5
Run 2	-6.5	-7.5	5.5	-4.5	1.5	2.5
Run 3	-5.5	4.5	-6.5	-7.5	2.5	-1.5
Run 4	-4.5	-5.5	-7.5	6.5	3.5	0.5
Run 5	-3.5	2.5	0.5	-1.5	-4.5	7.5
Run 6	-2.5	-3.5	1.5	0.5	-5.5	-6.5
Run 7	-1.5	0.5	-2.5	3.5	-6.5	5.5
Run 8	-0.5	-1.5	-3.5	-2.5	-7.5	-4.5
Run 9	7.5	-6.5	-4.5	-5.5	-0.5	3.5
Run 10	6.5	7.5	-5.5	4.5	-1.5	-2.5
Run 11	5.5	-4.5	6.5	7.5	-2.5	1.5
Run 12	4.5	5.5	7.5	-6.5	-3.5	-0.5
Run 13	3.5	-2.5	-0.5	1.5	4.5	-7.5
Run 14	2.5	3.5	-1.5	-0.5	5.5	6.5
Run 15	1.5	-0.5	2.5	-3.5	6.5	-5.5
Run 16	0.5	1.5	3.5	2.5	7.5	4.5

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

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-9	-9	-9	-8	2	7
Run 2	-8	5	5	4	-8	9
Run 3	-7	-6	-5	5	0	-5
Run 4	-6	-3	0	-9	-4	-2
Run 5	-5	9	6	1	7	4
Run 6	-4	7	9	-5	-2	-6
Run 7	-3	-2	3	3	-9	-7
Run 8	-2	6	-8	6	6	-3
Run 9	-1	-4	4	9	5	2
Run 10	0	4	-6	-2	9	-8
Run 11	1	0	-2	-4	-5	-1
Run 12	2	8	-3	-1	3	1
Run 13	3	-8	8	0	4	3
Run 14	4	-7	2	8	-1	-9
Run 15	5	-5	7	-3	8	5
Run 16	6	1	-4	7	-7	6
Run 17	7	2	-7	2	-3	8
Run 18	8	3	-1	-6	-6	-4
Run 19	9	-1	1	-7	1	0

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-9.5	6.5	-9.5	-8.5	-9.5	3.5
Run 2	-8.5	-6.5	4.5	8.5	-5.5	-1.5
Run 3	-7.5	-4.5	5.5	2.5	5.5	4.5
Run 4	-6.5	4.5	-1.5	-4.5	7.5	1.5
Run 5	-5.5	-7.5	0.5	-7.5	-6.5	-7.5
Run 6	-4.5	9.5	-0.5	5.5	6.5	-4.5
Run 7	-3.5	7.5	-4.5	4.5	-3.5	2.5
Run 8	-2.5	-1.5	3.5	-1.5	0.5	8.5
Run 9	-1.5	-9.5	-3.5	3.5	3.5	7.5
Run 10	-0.5	0.5	7.5	-5.5	-1.5	-6.5
Run 11	0.5	-0.5	8.5	-0.5	-0.5	-2.5
Run 12	1.5	-8.5	-5.5	6.5	4.5	-8.5
Run 13	2.5	2.5	-2.5	-6.5	1.5	-3.5
Run 14	3.5	5.5	1.5	1.5	9.5	-0.5
Run 15	4.5	1.5	-7.5	0.5	8.5	0.5
Run 16	5.5	8.5	9.5	7.5	-7.5	-5.5
Run 17	6.5	-2.5	-6.5	9.5	-8.5	6.5
Run 18	7.5	-5.5	6.5	-9.5	2.5	5.5
Run 19	8.5	-3.5	-8.5	-3.5	-2.5	-9.5
Run 20	9.5	3.5	2.5	-2.5	-4.5	9.5

Appendix II: Catalogue of Orthogonal Latin hypercube Designs of 2nd order with m (≤ 6) factors and n (≤ 20) runs

Number of factors =2

	Factor 1	Factor 2
Run 1	-1	0.5
Run 2	-0.5	-1
Run 3	0.5	1
Run 4	1	-0.5

	Factor 1	Factor 2
Run 1	-2	1
Run 2	-1	-2
Run 3	1	2
Run 4	2	-1
Run 5	0	0

	Factor 1	Factor 2
Run 1	-3	-2
Run 2	-2	3
Run 3	-1	0
Run 4	0	-1
Run 5	3	2
Run 6	2	-3
Run 7	1	0
Run 8	0	1

	Factor 1	Factor 2
Run 1	-4	-3
Run 2	-3	4
Run 3	-2	1
Run 4	-1	-2
Run 5	4	3
Run 6	3	-4
Run 7	2	-1
Run 8	1	2
Run 9	0	0

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

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

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

Number of factors =3

	Factor 1	Factor 2	Factor 3
Run 1	-3	-2	-1
Run 2	-2	3	0
Run 3	-1	0	3
Run 4	0	-1	2
Run 5	3	2	1
Run 6	2	-3	0
Run 7	1	0	-3
Run 8	0	1	-2

	Factor 1	Factor 2	Factor 3
Run 1	-4	-3	-2
Run 2	-3	4	-1
Run 3	-2	1	4
Run 4	-1	-2	3
Run 5	4	3	2
Run 6	3	-4	1
Run 7	2	-1	-4
Run 8	1	2	-3
Run 9	0	0	0

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

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

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

Number of factors =4

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-3	-2	-1	0
Run 2	-2	3	0	1
Run 3	-1	0	3	-2
Run 4	0	-1	2	3
Run 5	3	2	1	0
Run 6	2	-3	0	-1
Run 7	1	0	-3	2
Run 8	0	1	-2	-3

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-4	-3	-2	-1
Run 2	-3	4	-1	2
Run 3	-2	1	4	-3
Run 4	-1	-2	3	4
Run 5	4	3	2	1
Run 6	3	-4	1	-2
Run 7	2	-1	-4	3
Run 8	1	2	-3	-4
Run 9	0	0	0	0

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

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

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

Appendix III: Catalogue of Orthogonal Latin hypercube Designs of SFD with m (≤ 6) factors and n (≤ 20) runs

Number of factors =2

	Factor1	Factor2
Run1	-1	1
Run2	0	-1
Run3	1	0

	Factor1	Factor2
Run1	-1.5	-0.5
Run2	1.5	0.5
Run3	0.5	-1.5
Run4	-0.5	1.5

	Factor1	Factor2
Run1	-2	2
Run2	-1	0
Run3	0	-2
Run4	1	1
Run5	2	-1

	Factor1	Factor2
Run1	-3	3
Run2	-2	1
Run3	-1	-1
Run4	0	-3
Run5	1	2
Run6	2	0
Run7	3	-2

	Factor1	Factor2
Run1	-3.5	-2.5
Run2	-0.5	-1.5
Run3	0.5	1.5
Run4	3.5	2.5
Run5	-1.5	-3.5
Run6	-2.5	0.5
Run7	2.5	-0.5
Run8	1.5	3.5

	Factor1	Factor2
Run1	-4	4
Run2	-3	2
Run3	-2	0
Run4	-1	-2
Run5	0	-4
Run6	1	3
Run7	2	1
Run8	3	-1
Run9	4	-3

	Factor1	Factor2
Run1	-5	5
Run2	-4	3
Run3	-3	1
Run4	-2	-1
Run5	-1	-3
Run6	0	-5
Run7	1	4
Run8	2	2
Run9	3	0
Run10	4	-2
Run11	5	-4

Runs	Factor1	Factor2
Run1	-11	-9
Run2	-9	11
Run3	-7	-5
Run4	-5	7
Run5	-3	-1
Run6	-1	3
Run7	1	-3
Run8	3	1
Run9	5	-7
Run10	7	9
Run11	9	-11
Run12	11	9

Runs	Factor1	Factor2
Run1	-6	6
Run2	-5	4
Run3	-4	2
Run4	-3	0
Run5	-2	-2
Run6	-1	-4
Run7	0	-6
Run8	1	5
Run9	2	3
Run10	3	1
Run11	4	-1
Run12	5	-3
Run13	6	-5

Runs	Factor1	Factor2
Run1	-7	7
Run2	-6	5
Run3	-5	3
Run4	-4	1
Run5	-3	-1
Run6	-2	-3
Run7	-1	-5
Run8	0	-7
Run9	1	6
Run10	2	4
Run11	3	2
Run12	4	0
Run13	5	-2
Run14	6	-4
Run15	7	-6

Runs	Factor1	Factor2
Run1	-1	0
Run2	7	2
Run3	6	-2
Run4	3	1
Run5	-2	-4
Run6	8	6
Run7	-4	3
Run8	-5	8
Run9	4	7
Run10	0	5
Run11	1	-7
Run12	-6	-1
Run13	-7	-5
Run14	5	-6
Run15	-3	-8
Run16	2	-3
Run17	-8	4

Runs	Factor1	Factor2
Run1	-9	9
Run2	-8	7
Run3	-7	5
Run4	-6	3
Run5	-5	1
Run6	-4	-1
Run7	-3	-3
Run8	-2	-5
Run9	-1	-9
Run10	0	-5
Run11	1	8
Run12	2	6
Run13	3	4
Run14	4	2
Run15	5	0
Run16	6	-2
Run17	7	-4
Run18	8	-6
Run19	9	-8

Runs	Factor1	Factor2
Run1	-9.5	-8.5
Run2	-6.5	-7.5
Run3	-5.5	-4.5
Run4	-2.5	-3.5
Run5	-1.5	-0.5
Run6	1.5	0.5
Run7	2.5	3.5
Run8	5.5	4.5
Run9	6.5	7.5
Run10	9.5	8.5
Run11	-7.5	-9.5
Run12	-8.5	-5.5
Run13	-3.5	-6.5
Run14	-4.5	-1.5
Run15	0.5	-2.5
Run16	-0.5	2.5
Run17	4.5	1.5
Run18	3.5	6.5
Run19	8.5	5.5
Run20	7.5	9.5

Number of factors =3

Runs	Factor1	Factor2	Factor3
Run1	2	3	2
Run2	-2	1	3
Run3	1	2	-2
Run4	3	-1	0
Run5	0	-2	-3
Run6	-1	-3	1
Run7	-3	0	-1

Runs	Factor1	Factor2	Factor3
Run1	3.5	0.5	0.5
Run2	-2.5	-2.5	3.5
Run3	-3.5	-0.5	-0.5
Run4	-1.5	3.5	-1.5
Run5	1.5	-3.5	1.5
Run	-0.5	1.5	2.5
Run7	0.5	-1.5	-2.5
Run8	2.5	2.5	-3.5

Runs	Factor1	Factor2	Factor3
Run1	2	1	-4
Run2	0	-3	-3
Run3	4	-2	0
Run4	-1	4	-1
Run5	-2	-4	1
Run6	3	3	2
Run7	1	-1	4
Run8	-4	0	-2
Run9	-3	2	3

Runs	Factor1	Factor2	Factor3
Run1	2	-4	-4
Run2	4	4	-2
Run3	-1	2	-5
Run4	-5	1	3
Run5	-4	-3	-3
Run6	-2	-5	2
Run7	0	0	0
Run8	5	-1	1
Run9	3	3	4
Run10	1	-2	5
Run11	-3	-5	-1

Runs	Factor1	Factor2	Factor3
Run 1	-11	3	-3
Run 2	9	-5	-9
Run 3	3	-9	7
Run 4	11	-3	3
Run 5	-3	7	-11
Run 6	-7	1	11
Run 7	7	9	-5
Run 8	-1	-11	-7
Run 9	1	-1	-1
Run 10	-9	-7	1
Run 11	5	5	9
Run 12	-5	11	5

Runs	Factor1	Factor2	Factor3
Run1	4	3	-6
Run2	1	2	5
Run3	5	-3	6
Run4	-6	-4	-2
Run5	-5	-1	4
Run6	-4	5	3
Run7	-2	0	-1
Run8	0	-5	2
Run9	6	1	1
Run10	2	6	0
Run11	-3	4	-5
Run12	3	-2	-3
Run13	-1	-6	-4

Runs	Factor1	Factor2	Factor3
Run1	7	5	3
Run2	1	-6	6
Run3	5	-7	-3
Run4	6	-2	2
Run5	-4	7	-1
Run6	2	1	7
Run7	-7	0	-2
Run8	-6	-3	4
Run9	4	6	-4
Run10	-2	3	-7
Run11	0	2	1
Run12	3	-1	-5
Run13	-1	-5	0
Run14	-5	4	5
Run15	-3	-4	-6

Runs	Factor1	Factor2	Factor3
Run1	-7.5	-5.5	-2.5
Run2	-0.5	0.5	1.5
Run3	7.5	3.5	0.5
Run4	-2.5	7.5	2.5
Run5	6.5	-3.5	-0.5
Run6	-6.5	1.5	-1.5
Run7	-4.5	-4.5	4.5
Run8	4.5	-0.5	-6.5
Run9	-5.5	2.5	5.5
Run10	1.5	-7.5	3.5
Run11	5.5	6.5	-5.5
Run12	2.5	4.5	6.5
Run13	0.5	-6.5	-3.5
Run14	-3.5	-1.5	-7.5
Run15	-1.5	5.5	-4.5
Run16	3.5	-2.5	7.5

Runs	Factor1	Factor2	Factor3
Run1	-1	0	3
Run2	-2	-7	2
Run3	5	-2	-1
Run4	4	-5	5
Run5	-5	1	-4
Run6	6	-1	-8
Run7	-3	-4	8
Run8	-6	8	-2
Run9	7	3	4
Run10	0	6	7
Run11	3	-8	-6
Run12	-7	2	6
Run13	-4	-6	-5
Run14	1	5	-7
Run15	-8	-3	1
Run16	2	7	0
Run17	8	4	-3

Runs	Factor1	Factor2	Factor3
Run1	0	5	5
Run2	8	-3	-5
Run3	-6	4	-9
Run4	-1	-6	-8
Run5	-5	-7	4
Run6	3	0	0
Run7	-7	8	6
Run8	2	-8	-1
Run9	5	9	1
Run10	9	-4	3
Run11	7	3	7
Run12	4	-5	9
Run13	1	1	-7
Run14	-4	-2	-2
Run15	-9	2	2
Run16	-8	-9	-4
Run17	-3	-1	8
Run18	-2	7	-3
Run19	6	6	-6

Runs	Factor1	Factor2	Factor3
Run1	-2.5	9.5	-5.5
Run2	9.5	-8.5	-4.5
Run3	2.5	-7.5	0.5
Run4	7.5	-1.5	1.5
Run5	4.5	4.5	-8.5
Run6	1.5	0.5	6.5
Run7	-3.5	-5.5	5.5
Run8	6.5	6.5	-0.5
Run9	-1.5	-9.5	-7.5
Run10	-6.5	-2.5	-9.5
Run11	0.5	1.5	-1.5
Run12	5.5	-6.5	7.5
Run13	-9.5	-0.5	3.5
Run14	-5.5	3.5	9.5
Run15	-8.5	7.5	2.5
Run16	8.5	5.5	8.5
Run17	-7.5	2.5	-3.5
Run18	-0.5	8.5	4.5
Run19	3.5	-3.5	-6.5
Run20	-4.5	-4.5	-2.5

Number of factors =4

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-2.5	-3.5	-1.5	-0.5
Run2	1.5	0.5	-2.5	-3.5
Run3	0.5	-1.5	3.5	-2.5
Run4	2.5	3.5	1.5	0.5
Run5	-0.5	1.5	-3.5	2.5
Run6	-3.5	2.5	0.5	-1.5
Run7	-1.5	-0.5	2.5	3.5
Run8	3.5	-2.5	-0.5	1.5

Runs	Factor1	Factor2	Factor3	Factor4
Run1	2	4	2	-2
Run2	4	0	0	3
Run3	0	-4	-3	1
Run4	-1	1	3	4
Run5	-4	3	-2	2
Run6	-3	-1	-1	-4
Run7	-2	-3	4	0
Run8	1	2	-4	-1
Run9	3	-2	1	-3

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-1	0.2	0.2	-0.4
Run2	-0.8	0	-0.6	1
Run3	-0.2	0.8	0.4	0.6
Run4	0.6	0.6	-0.8	0.8
Run5	-0.4	-0.6	1	0.4
Run6	0	1	-0.2	-0.8
Run6	0.4	-0.2	-1	-0.6
Run8	0.8	-0.8	0	0.2
Run9	0.2	-0.4	0.6	-1
Run10	-0.6	-1	-0.4	-0.2
Run11	1	0.4	0.8	0

Runs	Factor1	Factor2	Factor3	Factor4
Run 1	-7	-1	11	7
Run2	-3	-9	-5	9
Run 3	-9	-5	-1	-11
Run 4	1	-11	7	-3
Run 5	-11	1	-9	1
Run 6	-1	5	-3	11
Run 7	5	11	9	3
Run 8	11	-3	3	5
Run 9	9	3	5	-9
Run 10	7	7	-7	-1
Run 11	3	-7	-11	-7
Run 12	-5	9	1	-5

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-2	-1	-6	-2
Run2	6	0	4	0
Run3	5	-5	-5	1
Run4	3	2	-1	5
Run5	-3	6	-4	2
Run6	-1	1	6	3
Run7	1	-3	1	-6
Run8	-4	-2	-2	6
Run9	2	-6	2	4
Run10	-5	-4	3	-1
Run11	0	5	5	-4
Run12	-6	3	0	-5
Run13	4	4	-3	-3

Runs	Factor1	Factor2	Factor3	Factor4
Run1	0	-7	5	-2
Run2	4	5	-3	-3
Run3	6	1	-5	5
Run4	-3	0	6	6
Run5	1	2	7	-4
Run6	3	6	3	4
Run7	5	-2	4	3
Run8	7	-4	2	-5
Run9	2	-6	-4	-1
Run10	-5	7	0	-7
Run11	-4	-1	-6	-6
Run12	-7	-5	-1	0
Run13	-1	-3	-2	7
Run14	-6	4	1	1
Run15	-2	3	-7	2

Runs	Factor1	Factor2	Factor3	Factor4
Run1	1.5	5.5	-5.5	-3.5
Run2	3.5	6.5	3.5	-5.5
Run3	-6.5	7.5	1.5	0.5
Run4	0.5	-5.5	6.5	1.5
Run5	6.5	-3.5	2.5	-4.5
Run6	-7.5	-1.5	4.5	5.5
Run7	4.5	-4.5	-1.5	6.5
Run8	-2.5	2.5	-4.5	7.5
Run9	-3.5	-7.5	-3.5	3.5
Run10	-4.5	-0.5	-7.5	-1.5
Run11	2.5	3.5	7.5	4.5
Run12	-5.5	1.5	5.5	-7.5
Run13	5.5	-2.5	-6.5	-2.5
Run14	7.5	4.5	-0.5	2.5
Run15	-0.5	0.5	0.5	-0.5
Run16	-1.5	-6.5	-2.5	-6.5

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-4	7	-6	2
Run2	-3	4	-4	-8
Run3	7	3	2	8
Run4	-5	-4	-7	6
Run5	-2	-3	8	5
Run6	-6	1	5	-3
Run7	8	-2	7	0
Run8	0	0	-1	1
Run9	-8	2	1	7
Run10	-1	8	3	3
Run11	4	-8	0	4
Run12	2	-5	-8	-2
Run13	6	6	-5	-1
Run14	-7	-6	-3	-4
Run15	5	-1	-2	-7
Run16	1	-7	4	-5
Run17	3	5	6	-6

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-4	5	-2	-8
Run2	4	2	-8	-3
Run3	6	-1	9	1
Run4	-9	1	-7	-1
Run5	-6	-8	-3	3
Run6	-1	-6	-5	-7
Run7	0	0	7	-9
Run8	2	-3	0	7
Run9	-8	-4	3	-6
Run10	9	-2	1	-5
Run11	-3	3	5	0
Run12	1	-9	4	-2
Run13	7	-7	-9	4
Run14	3	6	6	9
Run15	-7	4	-1	8
Run16	8	7	-4	5
Run17	-5	-5	8	6

Runs	Factor1	Factor2	Factor3	Factor4
Run1	-6.5	7.5	-4.5	-4.5
Run2	-3.5	-5.5	3.5	-9.5
Run3	-1.5	-8.5	-5.5	-3.5
Run4	-5.5	-9.5	2.5	4.5
Run5	8.5	-3.5	4.5	-6.5
Run6	-4.5	3.5	6.5	5.5
Run7	9.5	1.5	9.5	2.5
Run8	-0.5	-6.5	-7.5	8.5
Run9	5.5	8.5	-2.5	-8.5
Run10	-7.5	6.5	-3.5	6.5
Run11	6.5	-7.5	-0.5	3.5
Run12	2.5	-4.5	8.5	7.5
Run13	1.5	4.5	-9.5	-0.5
Run14	-9.5	-0.5	1.5	-2.5
Run15	3.5	9.5	5.5	0.5
Run16	7.5	-2.5	-6.5	-5.5
Run17	4.5	2.5	-1.5	9.5

Run18	-2	8	-6	2
Run19	5	9	2	-4

Run18	-8.5	-1.5	-8.5	1.5
Run19	-2.5	5.5	7.5	-7.5
Run20	0.5	0.5	0.5	-1.5

Number of factors =5

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	-2	0	2	-3	4
Run2	2	-3	-1	4	1
Run3	0	3	0	0	-4
Run4	4	-2	3	-2	-2
Run5	-1	-4	-2	-1	-3
Run6	3	4	1	1	3
Run7	-4	-1	4	3	-1
Run8	-3	2	-3	2	2
Run9	1	1	-4	-4	0

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	-3	3	1	0	5
Run2	2	5	2	-3	-1
Run3	5	-1	0	-2	4
Run4	0	-3	-5	4	2
Run5	-5	1	5	-1	-3
Run6	1	0	-4	-5	-2
Run7	3	2	4	5	1
Run8	-4	-4	-2	-4	3
Run9	4	-2	-1	1	-5
Run10	-1	-5	3	2	0
Run11	-2	4	-3	3	-4

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run 1	5	1	-7	-7	-11
Run 2	-9	-1	-3	-11	7
Run 3	-7	3	11	-1	9
Run 4	1	11	3	-9	1
Run 5	3	9	-5	5	11
Run 6	11	-3	-9	11	-7
Run 7	-3	-9	5	9	3
Run 8	9	7	7	7	-3
Run 9	7	-5	1	-5	5
Run 10	-1	-7	9	-3	-9
Run 11	-5	-11	-11	1	-1
Run 12	-11	5	-1	3	-5

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	-5	2	4	-4	0
Run2	5	5	2	-5	1
Run3	6	-2	0	0	5
Run4	-1	-5	3	-1	-5
Run5	4	0	-5	-6	-3
Run6	-2	6	-4	-3	4
Run7	-4	-4	-3	-2	3
Run8	3	1	6	2	-1
Run9	-3	-3	5	3	6
Run10	1	3	-2	6	2
Run11	-6	-1	1	5	-2
Run12	0	4	-1	1	-6
Run13	2	-6	-6	4	-4

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	5	-4	-5	-1	-5
Run2	-7	3	4	0	-4
Run3	2	6	-4	-4	-1

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	-4.5	6.5	7.5	-1.5	-3.5
Run2	-5.5	2.5	-3.5	4.5	5.5
Run3	-1.5	-7.5	3.5	0.5	-5.5

Run4	-1	-7	-3	5	4
Run5	-6	2	1	-6	5
Run6	3	-2	-1	-3	7
Run7	-5	-1	-7	1	-2
Run8	7	5	3	3	-3
Run9	4	0	5	-5	-6
Run10	-2	-6	0	-7	0
Run11	6	1	-6	6	2
Run12	1	7	7	-2	3
Run13	0	-3	6	2	1
Run14	-3	4	-2	4	6
Run15	-4	-5	2	7	-7

Run4	-7.5	-2.5	4.5	-0.5	2.5
Run5	3.5	3.5	2.5	6.5	-4.5
Run6	-3.5	0.5	-2.5	-7.5	-2.5
Run7	-2.5	4.5	0.5	-5.5	7.5
Run8	7.5	-5.5	6.5	-2.5	1.5
Run9	6.5	-0.5	-0.5	3.5	6.5
Run10	1.5	7.5	-7.5	1.5	0.5
Run11	5.5	5.5	1.5	-4.5	-0.5
Run12	4.5	-1.5	-4.5	-3.5	-7.5
Run13	-6.5	1.5	-1.5	2.5	-6.5
Run14	2.5	-4.5	-5.5	-6.5	4.5
Run15	-0.5	-6.5	5.5	7.5	3.5
Run16	0.5	-3.5	-6.5	5.5	-1.5

Runs	**Factor1**	**Factor2**	**Factor3**	**Factor4**	**Factor5**
Run1	6	-1	8	1	0
Run2	-5	-2	7	-4	4
Run3	-7	-3	-5	3	3
Run4	-4	-4	1	5	-8
Run5	2	-8	4	-8	-3
Run6	1	-7	2	6	2
Run7	3	8	3	-3	5
Run8	0	6	6	-1	-7
Run9	4	-6	-4	-2	6
Run10	8	4	-7	-5	1
Run11	-2	3	5	8	7
Run12	-3	1	-6	-6	8
Run13	-6	0	-3	-7	-4
Run14	5	5	-1	7	-1
Run15	-8	7	0	4	-2
Run16	-1	2	-8	2	-6
Run17	7	-5	-2	0	-5

Runs	**Factor1**	**Factor2**	**Factor3**	**Factor4**	**Factor5**
Run1	-2	8	-2	-5	-7
Run2	-5	-2	8	-6	-5
Run3	-6	-1	4	6	-4
Run4	-8	-9	1	-1	1
Run5	0	-3	-9	5	0
Run6	-4	9	9	1	-2
Run7	9	2	6	2	-6
Run8	2	-6	-1	0	-8
Run9	3	6	-3	7	-9
Run10	-9	1	2	4	9
Run11	-7	5	-8	-2	3
Run12	-3	-4	-7	-9	-3
Run13	7	4	-5	-4	2
Run14	5	3	7	3	7
Run15	8	-5	5	-7	-1
Run16	6	-8	-6	-3	6
Run17	1	-7	3	8	5
Run18	-1	0	0	-8	8
Run19	4	7	-4	9	4

Runs	Factor1	Factor2	Factor3	Factor4	Factor5
Run1	-1.5	0.5	-2.5	-5.5	-9.5
Run2	-5.5	-4.5	-3.5	5.5	-7.5
Run3	1.5	9.5	7.5	-6.5	-3.5
Run4	-3.5	4.5	6.5	4.5	2.5
Run5	-0.5	-7.5	-1.5	-9.5	1.5
Run6	8.5	-2.5	5.5	-1.5	-2.5
Run7	0.5	-5.5	9.5	-2.5	7.5
Run8	3.5	3.5	1.5	9.5	-6.5
Run9	5.5	-6.5	-5.5	3.5	9.5
Run10	9.5	5.5	2.5	6.5	5.5
Run11	-2.5	-1.5	8.5	1.5	-8.5
Run12	6.5	6.5	-6.5	-4.5	-1.5
Run13	-8.5	7.5	-4.5	2.5	-4.5
Run14	-9.5	-3.5	3.5	-3.5	-0.5
Run15	2.5	2.5	-9.5	7.5	0.5
Run16	4.5	-9.5	-8.5	0.5	-5.5
Run17	7.5	1.5	0.5	-7.5	8.5
Run18	-7.5	8.5	-0.5	-8.5	4.5
Run19	-6.5	-0.5	-7.5	-0.5	6.5
Run20	-4.5	-8.5	4.5	8.5	3.5

Number of factors =6

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-3	2	0	-4	-4	1
Run2	1	-3	-3	3	-1	-3
Run3	-1	1	-2	5	-5	5
Run4	4	4	4	1	-3	-2
Run5	-4	-2	-5	-1	3	3
Run6	-2	-4	5	-2	0	-5
Run7	0	-1	-1	-5	5	-4
Run8	5	-5	1	-3	-2	2
Run9	-5	3	2	4	1	-1
Run10	2	5	-4	0	2	0
Run11	3	0	3	2	4	4

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run 1	-11	-7	-7	3	-7	-3
Run 2	-9	7	7	9	-1	9
Run 3	9	5	9	-11	-3	5
Run 4	11	9	-1	-1	11	-1
Run 5	7	-5	-5	-5	-5	-11
Run 6	-7	11	-9	-3	1	-7
Run 7	-3	-9	11	1	-11	1
Run 8	5	3	-3	7	-9	3
Run 9	-1	1	3	11	5	-9
Run 10	3	-11	5	5	7	7
Run 11	1	-1	-11	-7	3	11
Run 12	-5	-3	1	-9	9	-5

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-6	0	4	-2	1	4
Run2	6	2	-4	3	-2	1
Run3	4	6	-2	-6	-1	-4
Run4	-5	3	0	2	2	-6
Run5	-4	1	-5	-1	-5	2
Run6	5	5	3	0	6	0
Run7	3	-3	-1	-5	0	5
Run8	2	-5	6	1	3	-2
Run9	-2	-4	-6	-4	5	-3
Run10	1	4	5	6	-3	-1
Run11	0	-2	2	-3	-6	-5
Run12	-1	-6	1	4	-4	6
Run13	-3	-1	-3	5	4	3

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	7	-7	-2	6	-2	4
Run2	0	6	6	1	-7	-4
Run3	1	3	-7	3	-5	-1
Run4	6	-1	5	-1	2	-6
Run5	-1	-3	7	5	1	7
Run6	-7	5	4	0	4	1
Run7	-2	-5	3	-7	5	2
Run8	-4	4	-1	-6	-1	-7
Run9	4	2	0	7	6	0
Run10	-3	1	-3	-5	-4	6
Run11	-6	-4	2	2	-3	-3
Run12	3	-6	-4	-4	-6	-2
Run13	5	7	1	-3	0	3
Run14	-5	0	-6	4	3	5
Run15	2	-2	-5	-2	7	-5

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-5.5	2.5	-5.5	7.5	-2.5	-2.5
Run2	3.5	1.5	6.5	-0.5	-6.5	1.5
Run3	-7.5	-7.5	4.5	2.5	6.5	-0.5
Run4	2.5	-6.5	1.5	5.5	1.5	4.5
Run5	-1.5	0.5	-7.5	-5.5	3.5	6.5
Run6	-6.5	-5.5	2.5	-3.5	-4.5	2.5
Run7	7.5	4.5	-2.5	3.5	0.5	5.5
Run8	4.5	-2.5	-4.5	4.5	-7.5	-1.5
Run9	-2.5	3.5	7.5	-1.5	5.5	3.5
Run10	1.5	-1.5	-3.5	-7.5	-5.5	-6.5
Run11	-4.5	5.5	-1.5	1.5	-3.5	7.5
Run12	5.5	-4.5	3.5	-6.5	2.5	0.5
Run13	6.5	6.5	-0.5	-4.5	7.5	-3.5
Run14	-0.5	-3.5	-6.5	0.5	4.5	-5.5
Run15	-3.5	7.5	0.5	-2.5	-1.5	-4.5
Run16	0.5	-0.5	5.5	6.5	-0.5	-7.5

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-3	-7	-2	-6	-2	2
Run2	-2	0	-7	8	-4	1
Run3	0	5	-3	-5	-8	-5
Run4	-8	3	-6	-2	5	-1
Run5	5	-5	6	4	-6	0
Run6	-6	-3	8	-3	-5	-6
Run7	-7	2	3	1	-3	7
Run8	6	6	1	-1	-7	6
Run9	-5	-8	2	6	2	-3
Run10	7	-4	-5	3	1	-7
Run11	8	-1	-8	-7	0	5
Run12	1	-2	5	-4	8	3
Run13	-4	4	4	7	7	-2
Run14	3	7	-4	2	6	4
Run15	2	8	7	0	-1	-4
Run16	4	-6	-1	5	3	8
Run17	-1	1	0	-8	4	-8

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-4	-3	4	-4	-7	-3
Run2	8	1	-4	2	-2	-9
Run3	-6	2	0	-8	-4	9
Run4	-5	-8	-5	7	-8	0
Run5	-9	7	-1	3	-1	1
Run6	0	-4	5	6	2	6
Run7	5	-5	3	-7	5	-1
Run8	1	9	7	-5	1	-5
Run9	2	-1	-9	4	0	8
Run10	-7	0	6	-6	9	4
Run11	4	8	2	-2	8	7
Run12	-3	4	-8	0	6	-8
Run13	3	5	-7	-9	-3	-2

Runs	Factor1	Factor2	Factor3	Factor4	Factor5	Factor6
Run1	-6.5	-9.5	6.5	-1.5	6.5	4.5
Run2	7.5	-6.5	-4.5	-6.5	-4.5	-3.5
Run3	9.5	5.5	-2.5	6.5	1.5	-5.5
Run4	-5.5	1.5	2.5	8.5	-8.5	-6.5
Run5	4.5	-2.5	-5.5	-4.5	9.5	-4.5
Run6	6.5	-4.5	9.5	-8.5	2.5	2.5
Run7	5.5	-7.5	4.5	9.5	-0.5	-0.5
Run8	-4.5	8.5	-8.5	-2.5	7.5	-7.5
Run9	-3.5	-8.5	-1.5	1.5	0.5	-8.5
Run10	-2.5	9.5	7.5	3.5	5.5	-2.5
Run11	1.5	0.5	-0.5	7.5	8.5	7.5
Run12	3.5	4.5	-3.5	-9.5	4.5	6.5
Run13	-1.5	-3.5	8.5	-3.5	-9.5	-1.5
Run14	-7.5	-0.5	5.5	5.5	-5.5	8.5
Run15	2.5	-5.5	-6.5	-0.5	-2.5	9.5

Run14	-2	-6	-2	9	7	-6
Run15	-1	3	9	8	-6	-4
Run16	6	6	1	1	-9	5
Run17	7	-2	8	5	4	-7
Run18	9	-9	-3	-1	-5	3
Run19	-8	-7	-6	-3	3	2

Run16	-0.5	3.5	1.5	-7.5	-1.5	-9.5
Run17	8.5	2.5	3.5	2.5	-6.5	5.5
Run18	-8.5	7.5	0.5	-5.5	-3.5	3.5
Run19	-9.5	-1.5	-7.5	0.5	3.5	1.5
Run20	0.5	6.5	-9.5	4.5	-7.5	0.5

Appendix III (b):-Catalogue of Orthogonal Latin hypercube Result of SFD with m (≤ 6) factors and n (≤ 20) runs

Factor	Runs	Orthogonality value	Space filling value		
			Phi Value	CL2 Value	Entropy Value
2	3	0.5	0.5001	0.2826	0.1553
	4	0	0.3658	0.1954	0.4588
	5	0.5	0.3713	0.1633	1.7735
	7	0.5	0.3712	0.1194	6.8841
	8	0.64	0.3663	0.1185	9.3637
	9	0.5	0.3795	0.0981	16.6201
	11	0.5	0.3859	0.0856	31.6612
	12	0.83	0.3669	0.1109	31.5279
	13	0.5	0.3911	0.0778	52.5075
	15	0.5	0.3954	0.0725	79.5574
	16	0.92	0.3678	0.109	89.0611
	17	0.5	0.3992	0.0688	113.143
	19	0.5	0.4026	0.066	153.549
	20	0.93	0.3681	0.1088	114.121
3	7	0.214;-0.071;0.285	0	0	0
	8	0.095;-0.285;-0.476	0	0	0
	9	0;0;0.083	0	0	0
	11	0.090;0.063;-0.036	0	0	0
	12	-0.104;-0.097;0.041	0.2511	0.1078	12.2306
	13	0.076;0.049;-0.032	0	0	0
	15	-0.05;0.028;-0.035	0	0	0
	16	0.120;-0.026;-0.032	0	0	0
	17	0.046;-0.203;0.031	0	0	0
	19	0.033;0.063;0.043	0	0	0
	20	-0.154;0.036;0.174	0	0	0
4	8	0.00; 0.00; 0.00; 0.00; 0.00; 0.00;	0.010021	0.123001	9.253
	9	0.00; 0.00; 0.00; 0.00; 0.00; 0.00	0.002134	0.110252	12.642
	11	-0.15; -0.12; 0.02; -0.12; 0.09; 0.20	0.001252	0.210350	16.358
	12	0.14;0.16;-0.02;0.13;0.07;0.09	0.012341	0.021035	25.146
	13	-0.03;p 0.10; 0.04; 0.09; -0.14; 0.07	0.101124	0.213056	39.152
	15	0.06; 0.16; -0.09; 0.12; -0.025; -0.10	0.120320	0.214541	66.254
	16	0.03;0.05; -0.01; -0.08; 0.02; 0.01	0.110325	0.154201	85.245
	17	0.00; 0.17; 0.01; -0.08; 0.02; 0.06	0.003215	0.012540	100.256

	19	0.11; 0.00; -0.04; 0.04; 0.07; 0.01	0.230126	0.251035	112.154
5	9	0.01; 0.00; -0.08; -0.20; -0.05; 0.15;0.12;0.06;-0.05; 0.11	0.001206	0.012543	11.586
	11	0.01; -0.08; 0.21; 0.10; 0.00; 0.11; -0.04; -0.17; 0.01; -0.07	0.012030	0.102013	19.254
	12	0.10;-0.14;0.28;-0.27;0.16;-0.11;0.20;-0.05;0.17;-0.08	0.023011	0.114578	29.354
	13	0.11; -0.16; 0.03; -0.18; -0.28; 0.07; -0.03; 0.11;0.13;-0.02	0.210013	0.221560	48.596
	15	0.11; -0.07; 0.22; 0.04; -0.13; -0.18; -0.07; 0.12; -0.13; -0.07	0.112057	0.215895	71.025
	16	-0.12; -0.07; -0.16; -0.05; -0.11; 0.06; 0.05; -0.01; -0.01; -0.01	0.210562	0.312546	94.058
	17	-0.06; 0.01; -0.01;-0.09; 0.11; 0.12; 0.01; 0.03; -0.01; -0.06	0.123594	0.356214	124.524
	19	0.01; -0.01; 0.04; 0.02; 0.18; 0.06; -0.06; -0.12; -0.08; 0.05	0.210025	0.415210	148.225
	20	-0.001; -0.03; 0.04; 0.04; -0.11; -0.06; 0.16;-0.07; 0.02;-0.10	0.013065	0.120546	156.235
6	11	-0.17; 0.18; 0.06; -0.05; 0.01; -0.01; -0.16; 0.01; 0.10; 0.11; 0.01; 0.16; -0.04; -0.01; -0.01	0.0125634	0.114523	23.654
	12	0.11;0.10;-0.28;0.17;0.06;-0.16;-0.04;0.23;-0.006;0.19;-0.08;0.23;-0.12;0.03;-0.02	0.021025	0.210356	37.695
	13	-0.12; -0.09; 0.06; -0.09; 0.07; 0.11; 0.01; -0.04; 0.28; -0.16; 0.13; -0.07; -0.07; 0.00; -0.03	0.245103	0.001253	62.126
	15	-0.17; -0.06; 0.12; 0.18; -0.07; 0.02; -0.02; -0.05; 0.08; 0.06; -0.08; -0.15; -0.02; 0.28; 0.06	0.350124	0.360128	92.698
	16	0.13; -0.05; -0.10; -0.11; -0.07; 0.01; -0.002; -0.02; 0.10; -0.13; -0.04; 0.04; 0.03; -0.002; 0.04	0.120589	0.264510	128.345
	17	0.02; -0.21; -0.02; -0.04; -0.11; 0.08; -0.11; -0.004; -0.08; 0.07; 0.20; -0.002; -0.16; 0.13; 0.02	0.010124	0.350102	182.366
	19	0.01; 0.06; 0.17; 0.04;n -0.24;; 0.02;-0.08; 0.03; 0.08; -0.09; -0.15; -0.01; -0.06; -0.21; -0.08	0.246531	0.353641	209.124
	20	-0.10; -0.02; -0.20; -0.06; 0.03; 0.05; 0.01; 0.02; -0.16; -0.15; -0.01; -0.12; 0.04; -0.02	0.451025	0.410203	318.235

Appendix IV: Catalogue of Nested Orthogonal Latin hypercube Designs (NOLH) with m (≤ 6) factors and p (≤6) layers

Number of Layers =2

	Factor 1	Factor 2
Run 1	-3	1
Run 2	-1	-3
Run 3	1	3
Run 4	3	-1
Run 5	-6	2
Run 6	-2	-6
Run 7	2	6
Run 8	6	-2
Run 9	0	0

Layer1- Run 1 to Run 9
Layer2- Run 1 to Run 4

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

Layer1- Run 1 to Run 15
Layer2- Run 1 to Run 7

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

Layer1- Run 1 to Run 17
Layer2- Run 1 to Run 8

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-10	-8	-10	-10	-6
Run 2	-8	4	-2	6	8
Run 3	-6	-4	8	10	-8
Run 4	-4	6	-6	8	2
Run 5	-2	8	4	-8	6
Run 6	0	-10	10	-4	10
Run 7	2	10	6	-6	-10
Run 8	4	-2	2	2	-4
Run 9	6	0	0	-2	0
Run 10	8	2	-8	0	4
Run 11	10	-6	-4	4	-2
Run 12	-11	-11	-3	-11	-7
Run 13	-9	-5	-5	11	9
Run 14	-7	9	11	-9	-1
Run 15	-5	1	1	1	1
Run 16	-3	5	-1	3	11
Run 17	-1	11	5	7	-5
Run 18	1	3	-11	5	-11
Run 19	3	-3	3	-3	3
Run 20	5	-9	7	9	-9
Run 21	7	-1	-9	-7	7
Run 22	9	7	-7	-5	-3
Run 23	11	-7	9	-1	5

Layer1- Run 1 to Run 23
Layer2- Run 1 to Run 11

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-10	-8	-10	-10	-6	0
Run 2	-8	4	-2	6	8	10
Run 3	-6	-4	8	10	-8	-4
Run 4	-4	6	-6	8	2	-8
Run 5	-2	8	4	-8	6	4
Run 6	0	-10	10	-4	10	-6
Run 7	2	10	6	-6	-10	-2
Run 8	4	-2	2	2	-4	6
Run 9	6	0	0	-2	0	2
Run 10	8	2	-8	0	4	-10
Run 11	10	-6	-4	4	-2	8
Run 12	-11	-11	-3	-11	-7	-7
Run 13	-9	-5	-5	11	9	1
Run 14	-7	9	11	-9	-1	3
Run 15	-5	1	1	1	1	11
Run 16	-3	5	-1	3	11	-9
Run 17	-1	11	5	7	-5	-3
Run 18	1	3	-11	5	-11	-5
Run 19	3	-3	3	-3	3	5
Run 20	5	-9	7	9	-9	7
Run 21	7	-1	-9	-7	7	9
Run 22	9	7	-7	-5	-3	-1
Run 23	11	-7	9	-1	5	-11

Layer1- Run 1 to Run 23
Layer2- Run 1 to Run 11

Number of Layers =3

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

Layer1- Run 1 to Run 17
Layer2- Run 1 to Run 9
Layer3- Run 1 to Run 5

	Factor 1	Factor 2	Factor 3
Run 1	-14	-10	-6
Run 2	-10	14	-2
Run 3	-6	2	14
Run 4	-2	-6	10
Run 5	14	10	6
Run 6	10	-14	2
Run 7	6	-2	-14
Run 8	2	6	-10
Run 9	-16	-8	0
Run 10	-12	16	8
Run 11	-8	-12	-16
Run 12	-4	12	-8
Run 13	0	-16	16
Run 14	4	8	-4
Run 15	8	0	12
Run 16	12	4	4
Run 17	16	-4	-12
Run 18	-15	13	9
Run 19	-13	-15	11
Run 20	-11	9	-13
Run 21	-9	-11	-15
Run 22	-7	5	1
Run 23	-5	-7	3
Run 24	-3	1	-5
Run 25	-1	-3	-7
Run 26	15	-13	-9
Run 27	13	15	-11
Run 28	11	-9	13
Run 29	9	11	15
Run 30	7	-5	-1
Run 31	5	7	-3
Run 32	3	-1	5
Run 33	1	3	7

Layer1- Run 1 to Run 33

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-14	-10	-6	-2
Run 2	-10	14	-2	6
Run 3	-6	2	14	-10
Run 4	-2	-6	10	14
Run 5	14	10	6	2
Run 6	10	-14	2	-6
Run 7	6	-2	-14	10
Run 8	2	6	-10	-14
Run 9	-16	-8	0	-12
Run 10	-12	16	8	4
Run 11	-8	-12	-16	-4
Run 12	-4	12	-8	12
Run 13	0	-16	16	16
Run 14	4	8	-4	0
Run 15	8	0	12	-8
Run 16	12	4	4	-16
Run 17	16	-4	-12	8
Run 18	-15	13	9	11
Run 19	-13	-15	11	-9
Run 20	-11	9	-13	-15
Run 21	-9	-11	-15	13
Run 22	-7	5	1	-3
Run 23	-5	-7	3	1
Run 24	-3	1	-5	7
Run 25	-1	-3	-7	-5
Run 26	15	-13	-9	-11
Run 27	13	15	-11	9
Run 28	11	-9	13	15
Run 29	9	11	15	-13
Run 30	7	-5	-1	3
Run 31	5	7	-3	-1
Run 32	3	-1	5	-7
Run 33	1	3	7	5

Layer1- Run 1 to Run 33

	Layer2- Run 1 to Run 17 **Layer3-** Run 1 to Run 9	**Layer2-** Run 1 to Run 17 **Layer3-** Run 1 to Run 9

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

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

Run 32	7	-3	-1	11	13
Run 33	-3	-1	7	13	9
Run 34	-1	7	-3	9	11
Run 35	-13	-11	-9	-3	7
Run 36	-11	-9	-13	7	-1
Run 37	-9	-13	-11	-1	-3
Run 38	-15	5	-19	-23	21
Run 39	-19	-15	5	21	-17
Run 40	5	-19	-15	-17	-23
Run 41	23	-21	17	-15	5
Run 42	-21	17	23	-19	-15
Run 43	17	23	-21	5	-19
Run 44	-7	3	1	-11	-13
Run 45	3	1	-7	-13	-9
Run 46	1	-7	3	-9	-11
Run 47	13	11	9	3	-7
Run 48	11	9	13	-7	1
Run 49	9	13	11	1	3

Layer1- Run 1 to Run 49
Layer2- Run 1 to Run 25
Layer3- Run 1 to Run 13

Run 32	7	-3	-1	11	13	9
Run 33	-3	-1	7	13	9	11
Run 34	-1	7	-3	9	11	13
Run 35	-13	-11	-9	-3	7	-1
Run 36	-11	-9	-13	7	-1	-3
Run 37	-9	-13	-11	-1	-3	7
Run 38	-15	5	-19	-23	21	-17
Run 39	-19	-15	5	21	-17	-23
Run 40	5	-19	-15	-17	-23	21
Run 41	23	-21	17	-15	5	-19
Run 42	-21	17	23	-19	-15	5
Run 43	17	23	-21	5	-19	-15
Run 44	-7	3	1	-11	-13	-9
Run 45	3	1	-7	-13	-9	-11
Run 46	1	-7	3	-9	-11	-13
Run 47	13	11	9	3	-7	1
Run 48	11	9	13	-7	1	3
Run 49	9	13	11	1	3	-7

Layer1- Run 1 to Run 49
Layer2- Run 1 to Run 25
Layer3- Run 1 to Run 13

Number of Layers =4

	Factor 1	**Factor 2**
Run 1	-12	4
Run 2	-4	-12
Run 3	4	12
Run 4	12	-4
Run 5	-24	8
Run 6	-8	-24
Run 7	8	24
Run 8	24	-8
Run 9	0	0
Run 10	-14	-10
Run 11	-10	14
Run 12	-6	2

	Factor 1	**Factor 2**	**Factor 3**
Run 1	-28	-20	-12
Run 2	-20	28	-4
Run 3	-12	4	28
Run 4	-4	-12	20
Run 5	28	20	12
Run 6	20	-28	4
Run 7	12	-4	-28
Run 8	4	12	-20
Run 9	-32	-16	0
Run 10	-24	32	16
Run 11	-16	-24	-32
Run 12	-8	24	-16

	Factor 1	**Factor 2**	**Factor 3**	**Factor 4**
Run 1	-28	-20	-12	-4
Run 2	-20	28	-4	12
Run 3	-12	4	28	-20
Run 4	-4	-12	20	28
Run 5	28	20	12	4
Run 6	20	-28	4	-12
Run 7	12	-4	-28	20
Run 8	4	12	-20	-28
Run 9	-32	-16	0	-24
Run 10	-24	32	16	8
Run 11	-16	-24	-32	-8
Run 12	-8	24	-16	24

Run 13	-2	-6
Run 14	14	10
Run 15	10	-14
Run 16	6	-2
Run 17	2	6
Run 18	-15	13
Run 19	-13	-15
Run 20	-11	9
Run 21	-9	-11
Run 22	-7	5
Run 23	-5	-7
Run 24	-3	1
Run 25	-1	-3
Run 26	15	-13
Run 27	13	15
Run 28	11	-9
Run 29	9	11
Run 30	7	-5
Run 31	5	7
Run 32	3	-1
Run 33	1	3

Layer1- Run 1 to Run 33
Layer2- Run 1 to Run 17

Layer3- Run 1 to Run 9
Layer4- Run 1 to Run 5

Run 13	0	-32	32
Run 14	8	16	-8
Run 15	16	0	24
Run 16	24	8	8
Run 17	32	-8	-24
Run 18	-30	26	18
Run 19	-26	-30	22
Run 20	-22	18	-26
Run 21	-18	-22	-30
Run 22	-14	10	2
Run 23	-10	-14	6
Run 24	-6	2	-10
Run 25	-2	-6	-14
Run 26	30	-26	-18
Run 27	26	30	-22
Run 28	22	-18	26
Run 29	18	22	30
Run 30	14	-10	-2
Run 31	10	14	-6
Run 32	6	-2	10
Run 33	2	6	14
Run 34	-31	29	25
Run 35	-29	-31	27
Run 36	-27	25	-29
Run 37	-25	-27	-31
Run 38	-23	21	17
Run 39	-21	-23	19
Run 40	-19	17	-21
Run 41	-17	-19	-23
Run 42	31	-29	-25
Run 43	29	31	-27
Run 44	27	-25	29
Run 45	25	27	31
Run 46	23	-21	-17
Run 47	21	23	-19
Run 48	19	-17	21
Run 49	17	19	23
Run 50	-15	13	9

Run 13	0	-32	32	32
Run 14	8	16	-8	0
Run 15	16	0	24	-16
Run 16	24	8	8	-32
Run 17	32	-8	-24	16
Run 18	-30	26	18	22
Run 19	-26	-30	22	-18
Run 20	-22	18	-26	-30
Run 21	-18	-22	-30	26
Run 22	-14	10	2	-6
Run 23	-10	-14	6	2
Run 24	-6	2	-10	14
Run 25	-2	-6	-14	-10
Run 26	30	-26	-18	-22
Run 27	26	30	-22	18
Run 28	22	-18	26	30
Run 29	18	22	30	-26
Run 30	14	-10	-2	6
Run 31	10	14	-6	-2
Run 32	6	-2	10	-14
Run 33	2	6	14	10
Run 34	-31	29	25	27
Run 35	-29	-31	27	-25
Run 36	-27	25	-29	-31
Run 37	-25	-27	-31	29
Run 38	-23	21	17	-19
Run 39	-21	-23	19	17
Run 40	-19	17	-21	23
Run 41	-17	-19	-23	-21
Run 42	31	-29	-25	-27
Run 43	29	31	-27	25
Run 44	27	-25	29	31
Run 45	25	27	31	-29
Run 46	23	-21	-17	19
Run 47	21	23	-19	-17
Run 48	19	-17	21	-23
Run 49	17	19	23	21
Run 50	-15	13	9	11

Run 51	-13	-15	11
Run 52	-11	9	-13
Run 53	-9	-11	-15
Run 54	-7	5	1
Run 55	-5	-7	3
Run 56	-3	1	-5
Run 57	-1	-3	-7
Run 58	15	-13	-9
Run 59	13	15	-11
Run 60	11	-9	13
Run 61	9	11	15
Run 62	7	-5	-1
Run 63	5	7	-3
Run 64	3	-1	5
Run 65	1	3	7

Layer1- Run 1 to Run 65
Layer2- Run 1 to Run 33
Layer3- Run 1 to Run 17
Layer4- Run 1 to Run 9

Run 51	-13	-15	11	-9
Run 52	-11	9	-13	-15
Run 53	-9	-11	-15	13
Run 54	-7	5	1	-3
Run 55	-5	-7	3	1
Run 56	-3	1	-5	7
Run 57	-1	-3	-7	-5
Run 58	15	-13	-9	-11
Run 59	13	15	-11	9
Run 60	11	-9	13	15
Run 61	9	11	15	-13
Run 62	7	-5	-1	3
Run 63	5	7	-3	-1
Run 64	3	-1	5	-7
Run 65	1	3	7	5

Layer1- Run 1 to Run 65
Layer2- Run 1 to Run 33
Layer3- Run 1 to Run 17
Layer4- Run 1 to Run 9

	Factor 1	**Factor 2**	**Factor 3**	**Factor 4**	**Factor 5**
Run 1	-44	-44	-12	-44	-28
Run 2	-36	-20	-20	44	36
Run 3	-28	36	44	-36	-4
Run 4	-20	4	4	4	4
Run 5	-12	20	-4	12	44
Run 6	-4	44	20	28	-20
Run 7	4	12	-44	20	-44
Run 8	12	-12	12	-12	12
Run 9	20	-36	28	36	-36
Run 10	28	-4	-36	-28	28
Run 11	36	28	-28	-20	-12
Run 12	44	-28	36	-4	20

	Factor 1	**Factor 2**	**Factor 3**	**Factor 4**	**Factor 5**	**Factor 6**
Run 1	-44	-44	-12	-44	-28	-28
Run 2	-36	-20	-20	44	36	4
Run 3	-28	36	44	-36	-4	12
Run 4	-20	4	4	4	4	44
Run 5	-12	20	-4	12	44	-36
Run 6	-4	44	20	28	-20	-12
Run 7	4	12	-44	20	-44	-20
Run 8	12	-12	12	-12	12	20
Run 9	20	-36	28	36	-36	28
Run 10	28	-4	-36	-28	28	36
Run 11	36	28	-28	-20	-12	-4

Run 13	-48	-48	-48	0	-40
Run 14	-40	8	32	-8	48
Run 15	-32	48	-32	40	40
Run 16	-24	16	48	-32	-48
Run 17	-16	-16	16	16	-16
Run 18	-8	24	8	8	-24
Run 19	0	32	-16	-48	8
Run 20	8	-32	-40	-16	24
Run 21	16	-40	40	48	16
Run 22	24	-24	24	-40	32
Run 23	32	-8	0	32	-8
Run 24	40	40	-8	24	-32
Run 25	48	0	-24	-24	0
Run 26	30	-10	38	46	-42
Run 27	38	30	-10	-42	34
Run 28	-10	38	30	34	46
Run 29	-46	42	-34	30	-10
Run 30	42	-34	-46	38	30
Run 31	-34	-46	42	-10	38
Run 32	14	-6	-2	22	26
Run 33	-6	-2	14	26	18
Run 34	-2	14	-6	18	22
Run 35	-26	-22	-18	-6	14
Run 36	-22	-18	-26	14	-2
Run 37	-18	-26	-22	-2	-6
Run 38	-30	10	-38	-46	42
Run 39	-38	-30	10	42	-34
Run 40	10	-38	-30	-34	-46
Run 41	46	-42	34	-30	10
Run 42	-42	34	46	-38	-30
Run 43	34	46	-42	10	-38
Run 44	-14	6	2	-22	-26
Run 45	6	2	-14	-26	-18
Run 46	2	-14	6	-18	-22
Run 47	26	22	18	6	-14
Run 48	22	18	26	-14	2
Run 49	18	26	22	2	6
Run 50	-47	45	41	43	33

Run 12	44	-28	36	-4	20	-44
Run 13	-48	-48	-48	0	-40	-8
Run 14	-40	8	32	-8	48	40
Run 15	-32	48	-32	40	40	-16
Run 16	-24	16	48	-32	-48	16
Run 17	-16	-16	16	16	-16	-32
Run 18	-8	24	8	8	-24	24
Run 19	0	32	-16	-48	8	-40
Run 20	8	-32	-40	-16	24	32
Run 21	16	-40	40	48	16	8
Run 22	24	-24	24	-40	32	-48
Run 23	32	-8	0	32	-8	-24
Run 24	40	40	-8	24	-32	0
Run 25	48	0	-24	-24	0	48
Run 26	30	-10	38	46	-42	34
Run 27	38	30	-10	-42	34	46
Run 28	-10	38	30	34	46	-42
Run 29	-46	42	-34	30	-10	38
Run 30	42	-34	-46	38	30	-10
Run 31	-34	-46	42	-10	38	30
Run 32	14	-6	-2	22	26	18
Run 33	-6	-2	14	26	18	22
Run 34	-2	14	-6	18	22	26
Run 35	-26	-22	-18	-6	14	-2
Run 36	-22	-18	-26	14	-2	-6
Run 37	-18	-26	-22	-2	-6	14
Run 38	-30	10	-38	-46	42	-34
Run 39	-38	-30	10	42	-34	-46
Run 40	10	-38	-30	-34	-46	42
Run 41	46	-42	34	-30	10	-38
Run 42	-42	34	46	-38	-30	10
Run 43	34	46	-42	10	-38	-30
Run 44	-14	6	2	-22	-26	-18
Run 45	6	2	-14	-26	-18	-22
Run 46	2	-14	6	-18	-22	-26
Run 47	26	22	18	6	-14	2
Run 48	22	18	26	-14	2	6
Run 49	18	26	22	2	6	-14

Run 51	-45	-47	43	-41	35
Run 52	-43	41	-45	-47	37
Run 53	-41	-43	-47	45	39
Run 54	-39	37	33	-35	-41
Run 55	-37	-39	35	33	-43
Run 56	-35	33	-37	39	-45
Run 57	-33	-35	-39	-37	-47
Run 58	47	-45	-41	-43	-33
Run 59	45	47	-43	41	-35
Run 60	43	-41	45	47	-37
Run 61	41	43	47	-45	-39
Run 62	39	-37	-33	35	41
Run 63	37	39	-35	-33	43
Run 64	35	-33	37	-39	45
Run 65	33	35	39	37	47
Run 66	-31	29	25	27	17
Run 67	-29	-31	27	-25	19
Run 68	-27	25	-29	-31	21
Run 69	-25	-27	-31	29	23
Run 70	-23	21	17	-19	-25
Run 71	-21	-23	19	17	-27
Run 72	-19	17	-21	23	-29
Run 73	-17	-19	-23	-21	-31
Run 74	31	-29	-25	-27	-17
Run 75	29	31	-27	25	-19
Run 76	27	-25	29	31	-21
Run 77	25	27	31	-29	-23
Run 78	23	-21	-17	19	25
Run 79	21	23	-19	-17	27
Run 80	19	-17	21	-23	29
Run 81	17	19	23	21	31
Run 82	-15	13	9	11	1
Run 83	-13	-15	11	-9	3
Run 84	-11	9	-13	-15	5
Run 85	-9	-11	-15	13	7
Run 86	-7	5	1	-3	-9
Run 87	-5	-7	3	1	-11
Run 88	-3	1	-5	7	-13

Run 50	-47	45	41	43	33	-39
Run 51	-45	-47	43	-41	35	37
Run 52	-43	41	-45	-47	37	-35
Run 53	-41	-43	-47	45	39	33
Run 54	-39	37	33	-35	-41	47
Run 55	-37	-39	35	33	-43	-45
Run 56	-35	33	-37	39	-45	43
Run 57	-33	-35	-39	-37	-47	-41
Run 58	47	-45	-41	-43	-33	39
Run 59	45	47	-43	41	-35	-37
Run 60	43	-41	45	47	-37	35
Run 61	41	43	47	-45	-39	-33
Run 62	39	-37	-33	35	41	-47
Run 63	37	39	-35	-33	43	45
Run 64	35	-33	37	-39	45	-43
Run 65	33	35	39	37	47	41
Run 66	-31	29	25	27	17	-23
Run 67	-29	-31	27	-25	19	21
Run 68	-27	25	-29	-31	21	-19
Run 69	-25	-27	-31	29	23	17
Run 70	-23	21	17	-19	-25	31
Run 71	-21	-23	19	17	-27	-29
Run 72	-19	17	-21	23	-29	27
Run 73	-17	-19	-23	-21	-31	-25
Run 74	31	-29	-25	-27	-17	23
Run 75	29	31	-27	25	-19	-21
Run 76	27	-25	29	31	-21	19
Run 77	25	27	31	-29	-23	-17
Run 78	23	-21	-17	19	25	-31
Run 79	21	23	-19	-17	27	29
Run 80	19	-17	21	-23	29	-27
Run 81	17	19	23	21	31	25
Run 82	-15	13	9	11	1	-7
Run 83	-13	-15	11	-9	3	5
Run 84	-11	9	-13	-15	5	-3
Run 85	-9	-11	-15	13	7	1
Run 86	-7	5	1	-3	-9	15
Run 87	-5	-7	3	1	-11	-13

Run 89	-1	-3	-7	-5	-15
Run 90	15	-13	-9	-11	-1
Run 91	13	15	-11	9	-3
Run 92	11	-9	13	15	-5
Run 93	9	11	15	-13	-7
Run 94	7	-5	-1	3	9
Run 95	5	7	-3	-1	11
Run 96	3	-1	5	-7	13
Run 97	1	3	7	5	15

Layer1- Run 1 to Run 97
Layer2- Run 1 to Run 49
Layer3- Run 1 to Run 25
Layer4- Run 1 to Run 13

Run 88	-3	1	-5	7	-13	11
Run 89	-1	-3	-7	-5	-15	-9
Run 90	15	-13	-9	-11	-1	7
Run 91	13	15	-11	9	-3	-5
Run 92	11	-9	13	15	-5	3
Run 93	9	11	15	-13	-7	-1
Run 94	7	-5	-1	3	9	-15
Run 95	5	7	-3	-1	11	13
Run 96	3	-1	5	-7	13	-11
Run 97	1	3	7	5	15	9

Layer1- Run 1 to Run 97
Layer2- Run 1 to Run 49
Layer3- Run 1 to Run 25
Layer4- Run 1 to Run 13

Number of Layers =5

	Factor 1	Factor 2
Run 1	-24	8
Run 2	-8	-24
Run 3	8	24
Run 4	24	-8
Run 5	-48	16
Run 6	-16	-48
Run 7	16	48
Run 8	48	-16
Run 9	0	0
Run 10	-28	-20
Run 11	-20	28
Run 12	-12	4
Run 13	-4	-12
Run 14	28	20
Run 15	20	-28
Run 16	12	-4
Run 17	4	12
Run 18	-30	26
Run 19	-26	-30
Run 20	-22	18
Run 21	-18	-22
Run 22	-14	10
Run 23	-10	-14
Run 24	-6	2
Run 25	-2	-6
Run 26	30	-26
Run 27	26	30
Run 28	22	-18
Run 29	18	22
Run 30	14	-10
Run 31	10	14
Run 32	6	-2
Run 33	2	6
Run 34	-31	29
Run 35	-29	-31

	Factor 1	Factor 2	Factor 3
Run 1	-56	-40	-24
Run 2	-40	56	-8
Run 3	-24	8	56
Run 4	-8	-24	40
Run 5	56	40	24
Run 6	40	-56	8
Run 7	24	-8	-56
Run 8	8	24	-40
Run 9	-64	-32	0
Run 10	-48	64	32
Run 11	-32	-48	-64
Run 12	-16	48	-32
Run 13	0	-64	64
Run 14	16	32	-16
Run 15	32	0	48
Run 16	48	16	16
Run 17	64	-16	-48
Run 18	-60	52	36
Run 19	-52	-60	44
Run 20	-44	36	-52
Run 21	-36	-44	-60
Run 22	-28	20	4
Run 23	-20	-28	12
Run 24	-12	4	-20
Run 25	-4	-12	-28
Run 26	60	-52	-36
Run 27	52	60	-44
Run 28	44	-36	52
Run 29	36	44	60
Run 30	28	-20	-4
Run 31	20	28	-12
Run 32	12	-4	20
Run 33	4	12	28
Run 34	-62	58	50
Run 35	-58	-62	54

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-56	-40	-24	-8
Run 2	-40	56	-8	24
Run 3	-24	8	56	-40
Run 4	-8	-24	40	56
Run 5	56	40	24	8
Run 6	40	-56	8	-24
Run 7	24	-8	-56	40
Run 8	8	24	-40	-56
Run 9	-64	-32	0	-48
Run 10	-48	64	32	16
Run 11	-32	-48	-64	-16
Run 12	-16	48	-32	48
Run 13	0	-64	64	64
Run 14	16	32	-16	0
Run 15	32	0	48	-32
Run 16	48	16	16	-64
Run 17	64	-16	-48	32
Run 18	-60	52	36	44
Run 19	-52	-60	44	-36
Run 20	-44	36	-52	-60
Run 21	-36	-44	-60	52
Run 22	-28	20	4	-12
Run 23	-20	-28	12	4
Run 24	-12	4	-20	28
Run 25	-4	-12	-28	-20
Run 26	60	-52	-36	-44
Run 27	52	60	-44	36
Run 28	44	-36	52	60
Run 29	36	44	60	-52
Run 30	28	-20	-4	12
Run 31	20	28	-12	-4
Run 32	12	-4	20	-28
Run 33	4	12	28	20
Run 34	-62	58	50	54
Run 35	-58	-62	54	-50

Run 36	-27	25
Run 37	-25	-27
Run 38	-23	21
Run 39	-21	-23
Run 40	-19	17
Run 41	-17	-19
Run 42	31	-29
Run 43	29	31
Run 44	27	-25
Run 45	25	27
Run 46	23	-21
Run 47	21	23
Run 48	19	-17
Run 49	17	19
Run 50	-15	13
Run 51	-13	-15
Run 52	-11	9
Run 53	-9	-11
Run 54	-7	5
Run 55	-5	-7
Run 56	-3	1
Run 57	-1	-3
Run 58	15	-13
Run 59	13	15
Run 60	11	-9
Run 61	9	11
Run 62	7	-5
Run 63	5	7
Run 64	3	-1
Run 65	1	3

Layer1- Run 1 to Run 65
Layer2- Run 1 to Run 33
Layer3- Run 1 to Run 17
Layer4- Run 1 to Run 9
Layer5- Run 1 to Run 5

Run 36	-54	50	-58
Run 37	-50	-54	-62
Run 38	-46	42	34
Run 39	-42	-46	38
Run 40	-38	34	-42
Run 41	-34	-38	-46
Run 42	62	-58	-50
Run 43	58	62	-54
Run 44	54	-50	58
Run 45	50	54	62
Run 46	46	-42	-34
Run 47	42	46	-38
Run 48	38	-34	42
Run 49	34	38	46
Run 50	-30	26	18
Run 51	-26	-30	22
Run 52	-22	18	-26
Run 53	-18	-22	-30
Run 54	-14	10	2
Run 55	-10	-14	6
Run 56	-6	2	-10
Run 57	-2	-6	-14
Run 58	30	-26	-18
Run 59	26	30	-22
Run 60	22	-18	26
Run 61	18	22	30
Run 62	14	-10	-2
Run 63	10	14	-6
Run 64	6	-2	10
Run 65	2	6	14
Run 66	-63	61	57
Run 67	-61	-63	59
Run 68	-59	57	-61
Run 69	-57	-59	-63
Run 70	-55	53	49
Run 71	-53	-55	51
Run 72	-51	49	-53
Run 73	-49	-51	-55

Run 36	-54	50	-58	-62
Run 37	-50	-54	-62	58
Run 38	-46	42	34	-38
Run 39	-42	-46	38	34
Run 40	-38	34	-42	46
Run 41	-34	-38	-46	-42
Run 42	62	-58	-50	-54
Run 43	58	62	-54	50
Run 44	54	-50	58	62
Run 45	50	54	62	-58
Run 46	46	-42	-34	38
Run 47	42	46	-38	-34
Run 48	38	-34	42	-46
Run 49	34	38	46	42
Run 50	-30	26	18	22
Run 51	-26	-30	22	-18
Run 52	-22	18	-26	-30
Run 53	-18	-22	-30	26
Run 54	-14	10	2	-6
Run 55	-10	-14	6	2
Run 56	-6	2	-10	14
Run 57	-2	-6	-14	-10
Run 58	30	-26	-18	-22
Run 59	26	30	-22	18
Run 60	22	-18	26	30
Run 61	18	22	30	-26
Run 62	14	-10	-2	6
Run 63	10	14	-6	-2
Run 64	6	-2	10	-14
Run 65	2	6	14	10
Run 66	-63	61	57	59
Run 67	-61	-63	59	-57
Run 68	-59	57	-61	-63
Run 69	-57	-59	-63	61
Run 70	-55	53	49	-51
Run 71	-53	-55	51	49
Run 72	-51	49	-53	55
Run 73	-49	-51	-55	-53

Run 74	63	-61	-57
Run 75	61	63	-59
Run 76	59	-57	61
Run 77	57	59	63
Run 78	55	-53	-49
Run 79	53	55	-51
Run 80	51	-49	53
Run 81	49	51	55
Run 82	-47	45	41
Run 83	-45	-47	43
Run 84	-43	41	-45
Run 85	-41	-43	-47
Run 86	-39	37	33
Run 87	-37	-39	35
Run 88	-35	33	-37
Run 89	-33	-35	-39
Run 90	47	-45	-41
Run 91	45	47	-43
Run 92	43	-41	45
Run 93	41	43	47
Run 94	39	-37	-33
Run 95	37	39	-35
Run 96	35	-33	37
Run 97	33	35	39
Run 98	-31	29	25
Run 99	-29	-31	27
Run 100	-27	25	-29
Run 101	-25	-27	-31
Run 102	-23	21	17
Run 103	-21	-23	19
Run 104	-19	17	-21
Run 105	-17	-19	-23
Run 106	31	-29	-25
Run 107	29	31	-27
Run 108	27	-25	29
Run 109	25	27	31
Run 110	23	-21	-17
Run 111	21	23	-19

Run 74	63	-61	-57	-59
Run 75	61	63	-59	57
Run 76	59	-57	61	63
Run 77	57	59	63	-61
Run 78	55	-53	-49	51
Run 79	53	55	-51	-49
Run 80	51	-49	53	-55
Run 81	49	51	55	53
Run 82	-47	45	41	43
Run 83	-45	-47	43	-41
Run 84	-43	41	-45	-47
Run 85	-41	-43	-47	45
Run 86	-39	37	33	-35
Run 87	-37	-39	35	33
Run 88	-35	33	-37	39
Run 89	-33	-35	-39	-37
Run 90	47	-45	-41	-43
Run 91	45	47	-43	41
Run 92	43	-41	45	47
Run 93	41	43	47	-45
Run 94	39	-37	-33	35
Run 95	37	39	-35	-33
Run 96	35	-33	37	-39
Run 97	33	35	39	37
Run 98	-31	29	25	27
Run 99	-29	-31	27	-25
Run 100	-27	25	-29	-31
Run 101	-25	-27	-31	29
Run 102	-23	21	17	-19
Run 103	-21	-23	19	17
Run 104	-19	17	-21	23
Run 105	-17	-19	-23	-21
Run 106	31	-29	-25	-27
Run 107	29	31	-27	25
Run 108	27	-25	29	31
Run 109	25	27	31	-29
Run 110	23	-21	-17	19
Run 111	21	23	-19	-17

Run 112	19	-17	21
Run 113	17	19	23
Run 114	-15	13	9
Run 115	-13	-15	11
Run 116	-11	9	-13
Run 117	-9	-11	-15
Run 118	-7	5	1
Run 119	-5	-7	3
Run 120	-3	1	-5
Run 121	-1	-3	-7
Run 122	15	-13	-9
Run 123	13	15	-11
Run 124	11	-9	13
Run 125	9	11	15
Run 126	7	-5	-1
Run 127	5	7	-3
Run 128	3	-1	5
Run 129	1	3	7

Layer1- Run 1 to Run 129
Layer2- Run 1 to Run 65
Layer3- Run 1 to Run 33
Layer4- Run 1 to Run 17
Layer5- Run 1 to Run 9

Run 112	19	-17	21	-23
Run 113	17	19	23	21
Run 114	-15	13	9	11
Run 115	-13	-15	11	-9
Run 116	-11	9	-13	-15
Run 117	-9	-11	-15	13
Run 118	-7	5	1	-3
Run 119	-5	-7	3	1
Run 120	-3	1	-5	7
Run 121	-1	-3	-7	-5
Run 122	15	-13	-9	-11
Run 123	13	15	-11	9
Run 124	11	-9	13	15
Run 125	9	11	15	-13
Run 126	7	-5	-1	3
Run 127	5	7	-3	-1
Run 128	3	-1	5	-7
Run 129	1	3	7	5

Layer1- Run 1 to Run 129
Layer2- Run 1 to Run 65
Layer3- Run 1 to Run 33
Layer4- Run 1 to Run 17
Layer5- Run 1 to Run 9

	Factor 1	**Factor 2**	**Factor 3**	**Factor 4**	**Factor 5**
Run 1	**-88**	**-88**	**-24**	**-88**	**-56**
Run 2	**-72**	**-40**	**-40**	**88**	**72**
Run 3	**-56**	**72**	**88**	**-72**	**-8**
Run 4	**-40**	**8**	**8**	**8**	**8**
Run 5	**-24**	**40**	**-8**	**24**	**88**
Run 6	**-8**	**88**	**40**	**56**	**-40**
Run 7	**8**	**24**	**-88**	**40**	**-88**
Run 8	**24**	**-24**	**24**	**-24**	**24**
Run 9	**40**	**-72**	**56**	**72**	**-72**
Run 10	**56**	**-8**	**-72**	**-56**	**56**
Run 11	**72**	**56**	**-56**	**-40**	**-24**

	Factor 1	**Factor 2**	**Factor 3**	**Factor 4**	**Factor 5**	**Factor 6**
Run 1	**-88**	**-88**	**-24**	**-88**	**-56**	**-56**
Run 2	**-72**	**-40**	**-40**	**88**	**72**	**8**
Run 3	**-56**	**72**	**88**	**-72**	**-8**	**24**
Run 4	**-40**	**8**	**8**	**8**	**8**	**88**
Run 5	**-24**	**40**	**-8**	**24**	**88**	**-72**
Run 6	**-8**	**88**	**40**	**56**	**-40**	**-24**
Run 7	**8**	**24**	**-88**	**40**	**-88**	**-40**
Run 8	**24**	**-24**	**24**	**-24**	**24**	**40**
Run 9	**40**	**-72**	**56**	**72**	**-72**	**56**
Run 10	**56**	**-8**	**-72**	**-56**	**56**	**72**
Run 11	**72**	**56**	**-56**	**-40**	**-24**	**-8**

Run 12	**88**	**-56**	**72**	**-8**	**40**
Run 13	**-96**	**-96**	**-96**	**0**	**-80**
Run 14	**-80**	**16**	**64**	**-16**	**96**
Run 15	**-64**	**96**	**-64**	**80**	**80**
Run 16	**-48**	**32**	**96**	**-64**	**-96**
Run 17	**-32**	**-32**	**32**	**32**	**-32**
Run 18	**-16**	**48**	**16**	**16**	**-48**
Run 19	**0**	**64**	**-32**	**-96**	**16**
Run 20	**16**	**-64**	**-80**	**-32**	**48**
Run 21	**32**	**-80**	**80**	**96**	**32**
Run 22	**48**	**-48**	**48**	**-80**	**64**
Run 23	**64**	**-16**	**0**	**64**	**-16**
Run 24	**80**	**80**	**-16**	**48**	**-64**
Run 25	**96**	**0**	**-48**	**-48**	**0**
Run 26	**60**	**-20**	**76**	**92**	**-84**
Run 27	**76**	**60**	**-20**	**-84**	**68**
Run 28	**-20**	**76**	**60**	**68**	**92**
Run 29	**-92**	**84**	**-68**	**60**	**-20**
Run 30	**84**	**-68**	**-92**	**76**	**60**
Run 31	**-68**	**-92**	**84**	**-20**	**76**
Run 32	**28**	**-12**	**-4**	**44**	**52**
Run 33	**-12**	**-4**	**28**	**52**	**36**
Run 34	**-4**	**28**	**-12**	**36**	**44**
Run 35	**-52**	**-44**	**-36**	**-12**	**28**
Run 36	**-44**	**-36**	**-52**	**28**	**-4**
Run 37	**-36**	**-52**	**-44**	**-4**	**-12**
Run 38	**-60**	**20**	**-76**	**-92**	**84**
Run 39	**-76**	**-60**	**20**	**84**	**-68**
Run 40	**20**	**-76**	**-60**	**-68**	**-92**
Run 41	**92**	**-84**	**68**	**-60**	**20**
Run 42	**-84**	**68**	**92**	**-76**	**-60**
Run 43	**68**	**92**	**-84**	**20**	**-76**
Run 44	**-28**	**12**	**4**	**-44**	**-52**
Run 45	**12**	**4**	**-28**	**-52**	**-36**
Run 46	**4**	**-28**	**12**	**-36**	**-44**
Run 47	**52**	**44**	**36**	**12**	**-28**
Run 48	**44**	**36**	**52**	**-28**	**4**
Run 49	**36**	**52**	**44**	**4**	**12**

Run 12	**88**	**-56**	**72**	**-8**	**40**	**-88**
Run 13	**-96**	**-96**	**-96**	**0**	**-80**	**-16**
Run 14	**-80**	**16**	**64**	**-16**	**96**	**80**
Run 15	**-64**	**96**	**-64**	**80**	**80**	**-32**
Run 16	**-48**	**32**	**96**	**-64**	**-96**	**32**
Run 17	**-32**	**-32**	**32**	**32**	**-32**	**-64**
Run 18	**-16**	**48**	**16**	**16**	**-48**	**48**
Run 19	**0**	**64**	**-32**	**-96**	**16**	**-80**
Run 20	**16**	**-64**	**-80**	**-32**	**48**	**64**
Run 21	**32**	**-80**	**80**	**96**	**32**	**16**
Run 22	**48**	**-48**	**48**	**-80**	**64**	**-96**
Run 23	**64**	**-16**	**0**	**64**	**-16**	**-48**
Run 24	**80**	**80**	**-16**	**48**	**-64**	**0**
Run 25	**96**	**0**	**-48**	**-48**	**0**	**96**
Run 26	**60**	**-20**	**76**	**92**	**-84**	**68**
Run 27	**76**	**60**	**-20**	**-84**	**68**	**92**
Run 28	**-20**	**76**	**60**	**68**	**92**	**-84**
Run 29	**-92**	**84**	**-68**	**60**	**-20**	**76**
Run 30	**84**	**-68**	**-92**	**76**	**60**	**-20**
Run 31	**-68**	**-92**	**84**	**-20**	**76**	**60**
Run 32	**28**	**-12**	**-4**	**44**	**52**	**36**
Run 33	**-12**	**-4**	**28**	**52**	**36**	**44**
Run 34	**-4**	**28**	**-12**	**36**	**44**	**52**
Run 35	**-52**	**-44**	**-36**	**-12**	**28**	**-4**
Run 36	**-44**	**-36**	**-52**	**28**	**-4**	**-12**
Run 37	**-36**	**-52**	**-44**	**-4**	**-12**	**28**
Run 38	**-60**	**20**	**-76**	**-92**	**84**	**-68**
Run 39	**-76**	**-60**	**20**	**84**	**-68**	**-92**
Run 40	**20**	**-76**	**-60**	**-68**	**-92**	**84**
Run 41	**92**	**-84**	**68**	**-60**	**20**	**-76**
Run 42	**-84**	**68**	**92**	**-76**	**-60**	**20**
Run 43	**68**	**92**	**-84**	**20**	**-76**	**-60**
Run 44	**-28**	**12**	**4**	**-44**	**-52**	**-36**
Run 45	**12**	**4**	**-28**	**-52**	**-36**	**-44**
Run 46	**4**	**-28**	**12**	**-36**	**-44**	**-52**
Run 47	**52**	**44**	**36**	**12**	**-28**	**4**
Run 48	**44**	**36**	**52**	**-28**	**4**	**12**
Run 49	**36**	**52**	**44**	**4**	**12**	**-28**

Run 50	**-94**	**90**	**82**	**86**	**66**
Run 51	**-90**	**-94**	**86**	**-82**	**70**
Run 52	**-86**	**82**	**-90**	**-94**	**74**
Run 53	**-82**	**-86**	**-94**	**90**	**78**
Run 54	**-78**	**74**	**66**	**-70**	**-82**
Run 55	**-74**	**-78**	**70**	**66**	**-86**
Run 56	**-70**	**66**	**-74**	**78**	**-90**
Run 57	**-66**	**-70**	**-78**	**-74**	**-94**
Run 58	**94**	**-90**	**-82**	**-86**	**-66**
Run 59	**90**	**94**	**-86**	**82**	**-70**
Run 60	**86**	**-82**	**90**	**94**	**-74**
Run 61	**82**	**86**	**94**	**-90**	**-78**
Run 62	**78**	**-74**	**-66**	**70**	**82**
Run 63	**74**	**78**	**-70**	**-66**	**86**
Run 64	**70**	**-66**	**74**	**-78**	**90**
Run 65	**66**	**70**	**78**	**74**	**94**
Run 66	**-62**	**58**	**50**	**54**	**34**
Run 67	**-58**	**-62**	**54**	**-50**	**38**
Run 68	**-54**	**50**	**-58**	**-62**	**42**
Run 69	**-50**	**-54**	**-62**	**58**	**46**
Run 70	**-46**	**42**	**34**	**-38**	**-50**
Run 71	**-42**	**-46**	**38**	**34**	**-54**
Run 72	**-38**	**34**	**-42**	**46**	**-58**
Run 73	**-34**	**-38**	**-46**	**-42**	**-62**
Run 74	**62**	**-58**	**-50**	**-54**	**-34**
Run 75	**58**	**62**	**-54**	**50**	**-38**
Run 76	**54**	**-50**	**58**	**62**	**-42**
Run 77	**50**	**54**	**62**	**-58**	**-46**
Run 78	**46**	**-42**	**-34**	**38**	**50**
Run 79	**42**	**46**	**-38**	**-34**	**54**
Run 80	**38**	**-34**	**42**	**-46**	**58**
Run 81	**34**	**38**	**46**	**42**	**62**
Run 82	**-30**	**26**	**18**	**22**	**2**
Run 83	**-26**	**-30**	**22**	**-18**	**6**
Run 84	**-22**	**18**	**-26**	**-30**	**10**
Run 85	**-18**	**-22**	**-30**	**26**	**14**
Run 86	**-14**	**10**	**2**	**-6**	**-18**
Run 87	**-10**	**-14**	**6**	**2**	**-22**

Run 50	**-94**	**90**	**82**	**86**	**66**	**-78**
Run 51	**-90**	**-94**	**86**	**-82**	**70**	**74**
Run 52	**-86**	**82**	**-90**	**-94**	**74**	**-70**
Run 53	**-82**	**-86**	**-94**	**90**	**78**	**66**
Run 54	**-78**	**74**	**66**	**-70**	**-82**	**94**
Run 55	**-74**	**-78**	**70**	**66**	**-86**	**-90**
Run 56	**-70**	**66**	**-74**	**78**	**-90**	**86**
Run 57	**-66**	**-70**	**-78**	**-74**	**-94**	**-82**
Run 58	**94**	**-90**	**-82**	**-86**	**-66**	**78**
Run 59	**90**	**94**	**-86**	**82**	**-70**	**-74**
Run 60	**86**	**-82**	**90**	**94**	**-74**	**70**
Run 61	**82**	**86**	**94**	**-90**	**-78**	**-66**
Run 62	**78**	**-74**	**-66**	**70**	**82**	**-94**
Run 63	**74**	**78**	**-70**	**-66**	**86**	**90**
Run 64	**70**	**-66**	**74**	**-78**	**90**	**-86**
Run 65	**66**	**70**	**78**	**74**	**94**	**82**
Run 66	**-62**	**58**	**50**	**54**	**34**	**-46**
Run 67	**-58**	**-62**	**54**	**-50**	**38**	**42**
Run 68	**-54**	**50**	**-58**	**-62**	**42**	**-38**
Run 69	**-50**	**-54**	**-62**	**58**	**46**	**34**
Run 70	**-46**	**42**	**34**	**-38**	**-50**	**62**
Run 71	**-42**	**-46**	**38**	**34**	**-54**	**-58**
Run 72	**-38**	**34**	**-42**	**46**	**-58**	**54**
Run 73	**-34**	**-38**	**-46**	**-42**	**-62**	**-50**
Run 74	**62**	**-58**	**-50**	**-54**	**-34**	**46**
Run 75	**58**	**62**	**-54**	**50**	**-38**	**-42**
Run 76	**54**	**-50**	**58**	**62**	**-42**	**38**
Run 77	**50**	**54**	**62**	**-58**	**-46**	**-34**
Run 78	**46**	**-42**	**-34**	**38**	**50**	**-62**
Run 79	**42**	**46**	**-38**	**-34**	**54**	**58**
Run 80	**38**	**-34**	**42**	**-46**	**58**	**-54**
Run 81	**34**	**38**	**46**	**42**	**62**	**50**
Run 82	**-30**	**26**	**18**	**22**	**2**	**-14**
Run 83	**-26**	**-30**	**22**	**-18**	**6**	**10**
Run 84	**-22**	**18**	**-26**	**-30**	**10**	**-6**
Run 85	**-18**	**-22**	**-30**	**26**	**14**	**2**
Run 86	**-14**	**10**	**2**	**-6**	**-18**	**30**
Run 87	**-10**	**-14**	**6**	**2**	**-22**	**-26**

Run 88	**-6**	**2**	**-10**	**14**	**-26**
Run 89	**-2**	**-6**	**-14**	**-10**	**-30**
Run 90	**30**	**-26**	**-18**	**-22**	**-2**
Run 91	**26**	**30**	**-22**	**18**	**-6**
Run 92	**22**	**-18**	**26**	**30**	**-10**
Run 93	**18**	**22**	**30**	**-26**	**-14**
Run 94	**14**	**-10**	**-2**	**6**	**18**
Run 95	**10**	**14**	**-6**	**-2**	**22**
Run 96	**6**	**-2**	**10**	**-14**	**26**
Run 97	**2**	**6**	**14**	**10**	**30**
Run 98	**-95**	**93**	**89**	**91**	**81**
Run 99	**-93**	**-95**	**91**	**-89**	**83**
Run 100	**-91**	**89**	**-93**	**-95**	**85**
Run 101	**-89**	**-91**	**-95**	**93**	**87**
Run 102	**-87**	**85**	**81**	**-83**	**-89**
Run 103	**-85**	**-87**	**83**	**81**	**-91**
Run 104	**-83**	**81**	**-85**	**87**	**-93**
Run 105	**-81**	**-83**	**-87**	**-85**	**-95**
Run 106	**95**	**-93**	**-89**	**-91**	**-81**
Run 107	**93**	**95**	**-91**	**89**	**-83**
Run 108	**91**	**-89**	**93**	**95**	**-85**
Run 109	**89**	**91**	**95**	**-93**	**-87**
Run 110	**87**	**-85**	**-81**	**83**	**89**
Run 111	**85**	**87**	**-83**	**-81**	**91**
Run 112	**83**	**-81**	**85**	**-87**	**93**
Run 113	**81**	**83**	**87**	**85**	**95**
Run 114	**-79**	**77**	**73**	**75**	**65**
Run 115	**-77**	**-79**	**75**	**-73**	**67**
Run 116	**-75**	**73**	**-77**	**-79**	**69**
Run 117	**-73**	**-75**	**-79**	**77**	**71**
Run 118	**-71**	**69**	**65**	**-67**	**-73**
Run 119	**-69**	**-71**	**67**	**65**	**-75**
Run 120	**-67**	**65**	**-69**	**71**	**-77**
Run 121	**-65**	**-67**	**-71**	**-69**	**-79**
Run 122	**79**	**-77**	**-73**	**-75**	**-65**
Run 123	**77**	**79**	**-75**	**73**	**-67**
Run 124	**75**	**-73**	**77**	**79**	**-69**
Run 125	**73**	**75**	**79**	**-77**	**-71**

Run 88	**-6**	**2**	**-10**	**14**	**-26**	**22**
Run 89	**-2**	**-6**	**-14**	**-10**	**-30**	**-18**
Run 90	**30**	**-26**	**-18**	**-22**	**-2**	**14**
Run 91	**26**	**30**	**-22**	**18**	**-6**	**-10**
Run 92	**22**	**-18**	**26**	**30**	**-10**	**6**
Run 93	**18**	**22**	**30**	**-26**	**-14**	**-2**
Run 94	**14**	**-10**	**-2**	**6**	**18**	**-30**
Run 95	**10**	**14**	**-6**	**-2**	**22**	**26**
Run 96	**6**	**-2**	**10**	**-14**	**26**	**-22**
Run 97	**2**	**6**	**14**	**10**	**30**	**18**
Run 98	**-95**	**93**	**89**	**91**	**81**	**-87**
Run 99	**-93**	**-95**	**91**	**-89**	**83**	**85**
Run 100	**-91**	**89**	**-93**	**-95**	**85**	**-83**
Run 101	**-89**	**-91**	**-95**	**93**	**87**	**81**
Run 102	**-87**	**85**	**81**	**-83**	**-89**	**95**
Run 103	**-85**	**-87**	**83**	**81**	**-91**	**-93**
Run 104	**-83**	**81**	**-85**	**87**	**-93**	**91**
Run 105	**-81**	**-83**	**-87**	**-85**	**-95**	**-89**
Run 106	**95**	**-93**	**-89**	**-91**	**-81**	**87**
Run 107	**93**	**95**	**-91**	**89**	**-83**	**-85**
Run 108	**91**	**-89**	**93**	**95**	**-85**	**83**
Run 109	**89**	**91**	**95**	**-93**	**-87**	**-81**
Run 110	**87**	**-85**	**-81**	**83**	**89**	**-95**
Run 111	**85**	**87**	**-83**	**-81**	**91**	**93**
Run 112	**83**	**-81**	**85**	**-87**	**93**	**-91**
Run 113	**81**	**83**	**87**	**85**	**95**	**89**
Run 114	**-79**	**77**	**73**	**75**	**65**	**-71**
Run 115	**-77**	**-79**	**75**	**-73**	**67**	**69**
Run 116	**-75**	**73**	**-77**	**-79**	**69**	**-67**
Run 117	**-73**	**-75**	**-79**	**77**	**71**	**65**
Run 118	**-71**	**69**	**65**	**-67**	**-73**	**79**
Run 119	**-69**	**-71**	**67**	**65**	**-75**	**-77**
Run 120	**-67**	**65**	**-69**	**71**	**-77**	**75**
Run 121	**-65**	**-67**	**-71**	**-69**	**-79**	**-73**
Run 122	**79**	**-77**	**-73**	**-75**	**-65**	**71**
Run 123	**77**	**79**	**-75**	**73**	**-67**	**-69**
Run 124	**75**	**-73**	**77**	**79**	**-69**	**67**
Run 125	**73**	**75**	**79**	**-77**	**-71**	**-65**

Run 126	**71**	**-69**	**-65**	**67**	**73**
Run 127	**69**	**71**	**-67**	**-65**	**75**
Run 128	**67**	**-65**	**69**	**-71**	**77**
Run 129	**65**	**67**	**71**	**69**	**79**
Run 130	**-63**	**61**	**57**	**59**	**49**
Run 131	**-61**	**-63**	**59**	**-57**	**51**
Run 132	**-59**	**57**	**-61**	**-63**	**53**
Run 133	**-57**	**-59**	**-63**	**61**	**55**
Run 134	**-55**	**53**	**49**	**-51**	**-57**
Run 135	**-53**	**-55**	**51**	**49**	**-59**
Run 136	**-51**	**49**	**-53**	**55**	**-61**
Run 137	**-49**	**-51**	**-55**	**-53**	**-63**
Run 138	**63**	**-61**	**-57**	**-59**	**-49**
Run 139	**61**	**63**	**-59**	**57**	**-51**
Run 140	**59**	**-57**	**61**	**63**	**-53**
Run 141	**57**	**59**	**63**	**-61**	**-55**
Run 142	**55**	**-53**	**-49**	**51**	**57**
Run 143	**53**	**55**	**-51**	**-49**	**59**
Run 144	**51**	**-49**	**53**	**-55**	**61**
Run 145	**49**	**51**	**55**	**53**	**63**
Run 146	**-47**	**45**	**41**	**43**	**33**
Run 147	**-45**	**-47**	**43**	**-41**	**35**
Run 148	**-43**	**41**	**-45**	**-47**	**37**
Run 149	**-41**	**-43**	**-47**	**45**	**39**
Run 150	**-39**	**37**	**33**	**-35**	**-41**
Run 151	**-37**	**-39**	**35**	**33**	**-43**
Run 152	**-35**	**33**	**-37**	**39**	**-45**
Run 153	**-33**	**-35**	**-39**	**-37**	**-47**
Run 154	**47**	**-45**	**-41**	**-43**	**-33**
Run 155	**45**	**47**	**-43**	**41**	**-35**
Run 156	**43**	**-41**	**45**	**47**	**-37**
Run 157	**41**	**43**	**47**	**-45**	**-39**
Run 158	**39**	**-37**	**-33**	**35**	**41**
Run 159	**37**	**39**	**-35**	**-33**	**43**
Run 160	**35**	**-33**	**37**	**-39**	**45**
Run 161	**33**	**35**	**39**	**37**	**47**
Run 162	**-31**	**29**	**25**	**27**	**17**
Run 163	**-29**	**-31**	**27**	**-25**	**19**

Run 126	**71**	**-69**	**-65**	**67**	**73**	**-79**
Run 127	**69**	**71**	**-67**	**-65**	**75**	**77**
Run 128	**67**	**-65**	**69**	**-71**	**77**	**-75**
Run 129	**65**	**67**	**71**	**69**	**79**	**73**
Run 130	**-63**	**61**	**57**	**59**	**49**	**-55**
Run 131	**-61**	**-63**	**59**	**-57**	**51**	**53**
Run 132	**-59**	**57**	**-61**	**-63**	**53**	**-51**
Run 133	**-57**	**-59**	**-63**	**61**	**55**	**49**
Run 134	**-55**	**53**	**49**	**-51**	**-57**	**63**
Run 135	**-53**	**-55**	**51**	**49**	**-59**	**-61**
Run 136	**-51**	**49**	**-53**	**55**	**-61**	**59**
Run 137	**-49**	**-51**	**-55**	**-53**	**-63**	**-57**
Run 138	**63**	**-61**	**-57**	**-59**	**-49**	**55**
Run 139	**61**	**63**	**-59**	**57**	**-51**	**-53**
Run 140	**59**	**-57**	**61**	**63**	**-53**	**51**
Run 141	**57**	**59**	**63**	**-61**	**-55**	**-49**
Run 142	**55**	**-53**	**-49**	**51**	**57**	**-63**
Run 143	**53**	**55**	**-51**	**-49**	**59**	**61**
Run 144	**51**	**-49**	**53**	**-55**	**61**	**-59**
Run 145	**49**	**51**	**55**	**53**	**63**	**57**
Run 146	**-47**	**45**	**41**	**43**	**33**	**-39**
Run 147	**-45**	**-47**	**43**	**-41**	**35**	**37**
Run 148	**-43**	**41**	**-45**	**-47**	**37**	**-35**
Run 149	**-41**	**-43**	**-47**	**45**	**39**	**33**
Run 150	**-39**	**37**	**33**	**-35**	**-41**	**47**
Run 151	**-37**	**-39**	**35**	**33**	**-43**	**-45**
Run 152	**-35**	**33**	**-37**	**39**	**-45**	**43**
Run 153	**-33**	**-35**	**-39**	**-37**	**-47**	**-41**
Run 154	**47**	**-45**	**-41**	**-43**	**-33**	**39**
Run 155	**45**	**47**	**-43**	**41**	**-35**	**-37**
Run 156	**43**	**-41**	**45**	**47**	**-37**	**35**
Run 157	**41**	**43**	**47**	**-45**	**-39**	**-33**
Run 158	**39**	**-37**	**-33**	**35**	**41**	**-47**
Run 159	**37**	**39**	**-35**	**-33**	**43**	**45**
Run 160	**35**	**-33**	**37**	**-39**	**45**	**-43**
Run 161	**33**	**35**	**39**	**37**	**47**	**41**
Run 162	**-31**	**29**	**25**	**27**	**17**	**-23**
Run 163	**-29**	**-31**	**27**	**-25**	**19**	**21**

Run 164	**-27**	**25**	**-29**	**-31**	**21**
Run 165	**-25**	**-27**	**-31**	**29**	**23**
Run 166	**-23**	**21**	**17**	**-19**	**-25**
Run 167	**-21**	**-23**	**19**	**17**	**-27**
Run 168	**-19**	**17**	**-21**	**23**	**-29**
Run 169	**-17**	**-19**	**-23**	**-21**	**-31**
Run 170	**31**	**-29**	**-25**	**-27**	**-17**
Run 171	**29**	**31**	**-27**	**25**	**-19**
Run 172	**27**	**-25**	**29**	**31**	**-21**
Run 173	**25**	**27**	**31**	**-29**	**-23**
Run 174	**23**	**-21**	**-17**	**19**	**25**
Run 175	**21**	**23**	**-19**	**-17**	**27**
Run 176	**19**	**-17**	**21**	**-23**	**29**
Run 177	**17**	**19**	**23**	**21**	**31**
Run 178	**-15**	**13**	**9**	**11**	**1**
Run 179	**-13**	**-15**	**11**	**-9**	**3**
Run 180	**-11**	**9**	**-13**	**-15**	**5**
Run 181	**-9**	**-11**	**-15**	**13**	**7**
Run 182	**-7**	**5**	**1**	**-3**	**-9**
Run 183	**-5**	**-7**	**3**	**1**	**-11**
Run 184	**-3**	**1**	**-5**	**7**	**-13**
Run 185	**-1**	**-3**	**-7**	**-5**	**-15**
Run 186	**15**	**-13**	**-9**	**-11**	**-1**
Run 187	**13**	**15**	**-11**	**9**	**-3**
Run 188	**11**	**-9**	**13**	**15**	**-5**
Run 189	**9**	**11**	**15**	**-13**	**-7**
Run 190	**7**	**-5**	**-1**	**3**	**9**
Run 191	**5**	**7**	**-3**	**-1**	**11**
Run 192	**3**	**-1**	**5**	**-7**	**13**
Run 193	**1**	**3**	**7**	**5**	**15**

Layer1- Run 1 to Run 193
Layer2- Run 1 to Run 97
Layer3- Run 1 to Run 49
Layer4- Run 1 to Run 25
Layer5- Run 1 to Run 13

Run 164	**-27**	**25**	**-29**	**-31**	**21**	**-19**
Run 165	**-25**	**-27**	**-31**	**29**	**23**	**17**
Run 166	**-23**	**21**	**17**	**-19**	**-25**	**31**
Run 167	**-21**	**-23**	**19**	**17**	**-27**	**-29**
Run 168	**-19**	**17**	**-21**	**23**	**-29**	**27**
Run 169	**-17**	**-19**	**-23**	**-21**	**-31**	**-25**
Run 170	**31**	**-29**	**-25**	**-27**	**-17**	**23**
Run 171	**29**	**31**	**-27**	**25**	**-19**	**-21**
Run 172	**27**	**-25**	**29**	**31**	**-21**	**19**
Run 173	**25**	**27**	**31**	**-29**	**-23**	**-17**
Run 174	**23**	**-21**	**-17**	**19**	**25**	**-31**
Run 175	**21**	**23**	**-19**	**-17**	**27**	**29**
Run 176	**19**	**-17**	**21**	**-23**	**29**	**-27**
Run 177	**17**	**19**	**23**	**21**	**31**	**25**
Run 178	**-15**	**13**	**9**	**11**	**1**	**-7**
Run 179	**-13**	**-15**	**11**	**-9**	**3**	**5**
Run 180	**-11**	**9**	**-13**	**-15**	**5**	**-3**
Run 181	**-9**	**-11**	**-15**	**13**	**7**	**1**
Run 182	**-7**	**5**	**1**	**-3**	**-9**	**15**
Run 183	**-5**	**-7**	**3**	**1**	**-11**	**-13**
Run 184	**-3**	**1**	**-5**	**7**	**-13**	**11**
Run 185	**-1**	**-3**	**-7**	**-5**	**-15**	**-9**
Run 186	**15**	**-13**	**-9**	**-11**	**-1**	**7**
Run 187	**13**	**15**	**-11**	**9**	**-3**	**-5**
Run 188	**11**	**-9**	**13**	**15**	**-5**	**3**
Run 189	**9**	**11**	**15**	**-13**	**-7**	**-1**
Run 190	**7**	**-5**	**-1**	**3**	**9**	**-15**
Run 191	**5**	**7**	**-3**	**-1**	**11**	**13**
Run 192	**3**	**-1**	**5**	**-7**	**13**	**-11**
Run 193	**1**	**3**	**7**	**5**	**15**	**9**

Layer1- Run 1 to Run 193
Layer2- Run 1 to Run 97
Layer3- Run 1 to Run 49
Layer4- Run 1 to Run 25
Layer5- Run 1 to Run 13

Number of Layers =6

	Factor 1	Factor 2
Run 1	-48	16
Run 2	-16	-48
Run 3	16	48
Run 4	48	-16
Run 5	-96	32
Run 6	-32	-96
Run 7	32	96
Run 8	96	-32
Run 9	0	0
Run 10	-56	-40
Run 11	-40	56
Run 12	-24	8
Run 13	-8	-24
Run 14	56	40
Run 15	40	-56
Run 16	24	-8
Run 17	8	24
Run 18	-60	52
Run 19	-52	-60
Run 20	-44	36
Run 21	-36	-44
Run 22	-28	20
Run 23	-20	-28
Run 24	-12	4
Run 25	-4	-12
Run 26	60	-52
Run 27	52	60
Run 28	44	-36
Run 29	36	44
Run 30	28	-20
Run 31	20	28
Run 32	12	-4
Run 33	4	12

	Factor 1	Factor 2	Factor 3
Run 1	-112	-80	-48
Run 2	-80	112	-16
Run 3	-48	16	112
Run 4	-16	-48	80
Run 5	112	80	48
Run 6	80	-112	16
Run 7	48	-16	-112
Run 8	16	48	-80
Run 9	-128	-64	0
Run 10	-96	128	64
Run 11	-64	-96	-128
Run 12	-32	96	-64
Run 13	0	-128	128
Run 14	32	64	-32
Run 15	64	0	96
Run 16	96	32	32
Run 17	128	-32	-96
Run 18	-120	104	72
Run 19	-104	-120	88
Run 20	-88	72	-104
Run 21	-72	-88	-120
Run 22	-56	40	8
Run 23	-40	-56	24
Run 24	-24	8	-40
Run 25	-8	-24	-56
Run 26	120	-104	-72
Run 27	104	120	-88
Run 28	88	-72	104
Run 29	72	88	120
Run 30	56	-40	-8
Run 31	40	56	-24
Run 32	24	-8	40
Run 33	8	24	56

	Factor 1	Factor 2	Factor 3	Factor 4
Run 1	-112	-80	-48	-16
Run 2	-80	112	-16	48
Run 3	-48	16	112	-80
Run 4	-16	-48	80	112
Run 5	112	80	48	16
Run 6	80	-112	16	-48
Run 7	48	-16	-112	80
Run 8	16	48	-80	-112
Run 9	-128	-64	0	-96
Run 10	-96	128	64	32
Run 11	-64	-96	-128	-32
Run 12	-32	96	-64	96
Run 13	0	-128	128	128
Run 14	32	64	-32	0
Run 15	64	0	96	-64
Run 16	96	32	32	-128
Run 17	128	-32	-96	64
Run 18	-120	104	72	88
Run 19	-104	-120	88	-72
Run 20	-88	72	-104	-120
Run 21	-72	-88	-120	104
Run 22	-56	40	8	-24
Run 23	-40	-56	24	8
Run 24	-24	8	-40	56
Run 25	-8	-24	-56	-40
Run 26	120	-104	-72	-88
Run 27	104	120	-88	72
Run 28	88	-72	104	120
Run 29	72	88	120	-104
Run 30	56	-40	-8	24
Run 31	40	56	-24	-8
Run 32	24	-8	40	-56
Run 33	8	24	56	40

Run 34	**-62**	**58**
Run 35	**-58**	**-62**
Run 36	**-54**	**50**
Run 37	**-50**	**-54**
Run 38	**-46**	**42**
Run 39	**-42**	**-46**
Run 40	**-38**	**34**
Run 41	**-34**	**-38**
Run 42	**62**	**-58**
Run 43	**58**	**62**
Run 44	**54**	**-50**
Run 45	**50**	**54**
Run 46	**46**	**-42**
Run 47	**42**	**46**
Run 48	**38**	**-34**
Run 49	**34**	**38**
Run 50	**-30**	**26**
Run 51	**-26**	**-30**
Run 52	**-22**	**18**
Run 53	**-18**	**-22**
Run 54	**-14**	**10**
Run 55	**-10**	**-14**
Run 56	**-6**	**2**
Run 57	**-2**	**-6**
Run 58	**30**	**-26**
Run 59	**26**	**30**
Run 60	**22**	**-18**
Run 61	**18**	**22**
Run 62	**14**	**-10**
Run 63	**10**	**14**
Run 64	**6**	**-2**
Run 65	**2**	**6**
Run 66	**-63**	**61**
Run 67	**-61**	**-63**
Run 68	**-59**	**57**
Run 69	**-57**	**-59**
Run 70	**-55**	**53**
Run 71	**-53**	**-55**

Run 34	**-124**	**116**	**100**
Run 35	**-116**	**-124**	**108**
Run 36	**-108**	**100**	**-116**
Run 37	**-100**	**-108**	**-124**
Run 38	**-92**	**84**	**68**
Run 39	**-84**	**-92**	**76**
Run 40	**-76**	**68**	**-84**
Run 41	**-68**	**-76**	**-92**
Run 42	**124**	**-116**	**-100**
Run 43	**116**	**124**	**-108**
Run 44	**108**	**-100**	**116**
Run 45	**100**	**108**	**124**
Run 46	**92**	**-84**	**-68**
Run 47	**84**	**92**	**-76**
Run 48	**76**	**-68**	**84**
Run 49	**68**	**76**	**92**
Run 50	**-60**	**52**	**36**
Run 51	**-52**	**-60**	**44**
Run 52	**-44**	**36**	**-52**
Run 53	**-36**	**-44**	**-60**
Run 54	**-28**	**20**	**4**
Run 55	**-20**	**-28**	**12**
Run 56	**-12**	**4**	**-20**
Run 57	**-4**	**-12**	**-28**
Run 58	**60**	**-52**	**-36**
Run 59	**52**	**60**	**-44**
Run 60	**44**	**-36**	**52**
Run 61	**36**	**44**	**60**
Run 62	**28**	**-20**	**-4**
Run 63	**20**	**28**	**-12**
Run 64	**12**	**-4**	**20**
Run 65	**4**	**12**	**28**
Run 66	**-126**	**122**	**114**
Run 67	**-122**	**-126**	**118**
Run 68	**-118**	**114**	**-122**
Run 69	**-114**	**-118**	**-126**
Run 70	**-110**	**106**	**98**
Run 71	**-106**	**-110**	**102**

Run 34	**-124**	**116**	**100**	**108**
Run 35	**-116**	**-124**	**108**	**-100**
Run 36	**-108**	**100**	**-116**	**-124**
Run 37	**-100**	**-108**	**-124**	**116**
Run 38	**-92**	**84**	**68**	**-76**
Run 39	**-84**	**-92**	**76**	**68**
Run 40	**-76**	**68**	**-84**	**92**
Run 41	**-68**	**-76**	**-92**	**-84**
Run 42	**124**	**-116**	**-100**	**-108**
Run 43	**116**	**124**	**-108**	**100**
Run 44	**108**	**-100**	**116**	**124**
Run 45	**100**	**108**	**124**	**-116**
Run 46	**92**	**-84**	**-68**	**76**
Run 47	**84**	**92**	**-76**	**-68**
Run 48	**76**	**-68**	**84**	**-92**
Run 49	**68**	**76**	**92**	**84**
Run 50	**-60**	**52**	**36**	**44**
Run 51	**-52**	**-60**	**44**	**-36**
Run 52	**-44**	**36**	**-52**	**-60**
Run 53	**-36**	**-44**	**-60**	**52**
Run 54	**-28**	**20**	**4**	**-12**
Run 55	**-20**	**-28**	**12**	**4**
Run 56	**-12**	**4**	**-20**	**28**
Run 57	**-4**	**-12**	**-28**	**-20**
Run 58	**60**	**-52**	**-36**	**-44**
Run 59	**52**	**60**	**-44**	**36**
Run 60	**44**	**-36**	**52**	**60**
Run 61	**36**	**44**	**60**	**-52**
Run 62	**28**	**-20**	**-4**	**12**
Run 63	**20**	**28**	**-12**	**-4**
Run 64	**12**	**-4**	**20**	**-28**
Run 65	**4**	**12**	**28**	**20**
Run 66	**-126**	**122**	**114**	**118**
Run 67	**-122**	**-126**	**118**	**-114**
Run 68	**-118**	**114**	**-122**	**-126**
Run 69	**-114**	**-118**	**-126**	**122**
Run 70	**-110**	**106**	**98**	**-102**
Run 71	**-106**	**-110**	**102**	**98**

Run 72	-51	49
Run 73	-49	-51
Run 74	63	-61
Run 75	61	63
Run 76	59	-57
Run 77	57	59
Run 78	55	-53
Run 79	53	55
Run 80	51	-49
Run 81	49	51
Run 82	-47	45
Run 83	-45	-47
Run 84	-43	41
Run 85	-41	-43
Run 86	-39	37
Run 87	-37	-39
Run 88	-35	33
Run 89	-33	-35
Run 90	47	-45
Run 91	45	47
Run 92	43	-41
Run 93	41	43
Run 94	39	-37
Run 95	37	39
Run 96	35	-33
Run 97	33	35
Run 98	-31	29
Run 99	-29	-31
Run 100	-27	25
Run 101	-25	-27
Run 102	-23	21
Run 103	-21	-23
Run 104	-19	17
Run 105	-17	-19
Run 106	31	-29
Run 107	29	31
Run 108	27	-25
Run 109	25	27

Run 72	-102	98	-106
Run 73	-98	-102	-110
Run 74	126	-122	-114
Run 75	122	126	-118
Run 76	118	-114	122
Run 77	114	118	126
Run 78	110	-106	-98
Run 79	106	110	-102
Run 80	102	-98	106
Run 81	98	102	110
Run 82	-94	90	82
Run 83	-90	-94	86
Run 84	-86	82	-90
Run 85	-82	-86	-94
Run 86	-78	74	66
Run 87	-74	-78	70
Run 88	-70	66	-74
Run 89	-66	-70	-78
Run 90	94	-90	-82
Run 91	90	94	-86
Run 92	86	-82	90
Run 93	82	86	94
Run 94	78	-74	-66
Run 95	74	78	-70
Run 96	70	-66	74
Run 97	66	70	78
Run 98	-62	58	50
Run 99	-58	-62	54
Run 100	-54	50	-58
Run 101	-50	-54	-62
Run 102	-46	42	34
Run 103	-42	-46	38
Run 104	-38	34	-42
Run 105	-34	-38	-46
Run 106	62	-58	-50
Run 107	58	62	-54
Run 108	54	-50	58
Run 109	50	54	62

Run 72	-102	98	-106	110
Run 73	-98	-102	-110	-106
Run 74	126	-122	-114	-118
Run 75	122	126	-118	114
Run 76	118	-114	122	126
Run 77	114	118	126	-122
Run 78	110	-106	-98	102
Run 79	106	110	-102	-98
Run 80	102	-98	106	-110
Run 81	98	102	110	106
Run 82	-94	90	82	86
Run 83	-90	-94	86	-82
Run 84	-86	82	-90	-94
Run 85	-82	-86	-94	90
Run 86	-78	74	66	-70
Run 87	-74	-78	70	66
Run 88	-70	66	-74	78
Run 89	-66	-70	-78	-74
Run 90	94	-90	-82	-86
Run 91	90	94	-86	82
Run 92	86	-82	90	94
Run 93	82	86	94	-90
Run 94	78	-74	-66	70
Run 95	74	78	-70	-66
Run 96	70	-66	74	-78
Run 97	66	70	78	74
Run 98	-62	58	50	54
Run 99	-58	-62	54	-50
Run 100	-54	50	-58	-62
Run 101	-50	-54	-62	58
Run 102	-46	42	34	-38
Run 103	-42	-46	38	34
Run 104	-38	34	-42	46
Run 105	-34	-38	-46	-42
Run 106	62	-58	-50	-54
Run 107	58	62	-54	50
Run 108	54	-50	58	62
Run 109	50	54	62	-58

Run 110	**23**	**-21**
Run 111	**21**	**23**
Run 112	**19**	**-17**
Run 113	**17**	**19**
Run 114	**-15**	**13**
Run 115	**-13**	**-15**
Run 116	**-11**	**9**
Run 117	**-9**	**-11**
Run 118	**-7**	**5**
Run 119	**-5**	**-7**
Run 120	**-3**	**1**
Run 121	**-1**	**-3**
Run 122	**15**	**-13**
Run 123	**13**	**15**
Run 124	**11**	**-9**
Run 125	**9**	**11**
Run 126	**7**	**-5**
Run 127	**5**	**7**
Run 128	**3**	**-1**
Run 129	**1**	**3**

Layer1- Run 1 to Run 129
Layer2- Run 1 to Run 65
Layer3- Run 1 to Run 33
Layer4- Run 1 to Run 17
Layer5- Run 1 to Run 9
Layer6- Run 1 to Run 5

Run 110	**46**	**-42**	**-34**
Run 111	**42**	**46**	**-38**
Run 112	**38**	**-34**	**42**
Run 113	**34**	**38**	**46**
Run 114	**-30**	**26**	**18**
Run 115	**-26**	**-30**	**22**
Run 116	**-22**	**18**	**-26**
Run 117	**-18**	**-22**	**-30**
Run 118	**-14**	**10**	**2**
Run 119	**-10**	**-14**	**6**
Run 120	**-6**	**2**	**-10**
Run 121	**-2**	**-6**	**-14**
Run 122	**30**	**-26**	**-18**
Run 123	**26**	**30**	**-22**
Run 124	**22**	**-18**	**26**
Run 125	**18**	**22**	**30**
Run 126	**14**	**-10**	**-2**
Run 127	**10**	**14**	**-6**
Run 128	**6**	**-2**	**10**
Run 129	**2**	**6**	**14**
Run 130	**-127**	**125**	**121**
Run 131	**-125**	**-127**	**123**
Run 132	**-123**	**121**	**-125**
Run 133	**-121**	**-123**	**-127**
Run 134	**-119**	**117**	**113**
Run 135	**-117**	**-119**	**115**
Run 136	**-115**	**113**	**-117**
Run 137	**-113**	**-115**	**-119**
Run 138	**127**	**-125**	**-121**
Run 139	**125**	**127**	**-123**
Run 140	**123**	**-121**	**125**
Run 141	**121**	**123**	**127**
Run 142	**119**	**-117**	**-113**
Run 143	**117**	**119**	**-115**
Run 144	**115**	**-113**	**117**
Run 145	**113**	**115**	**119**
Run 146	**-111**	**109**	**105**
Run 147	**-109**	**-111**	**107**

Run 110	**46**	**-42**	**-34**	**38**
Run 111	**42**	**46**	**-38**	**-34**
Run 112	**38**	**-34**	**42**	**-46**
Run 113	**34**	**38**	**46**	**42**
Run 114	**-30**	**26**	**18**	**22**
Run 115	**-26**	**-30**	**22**	**-18**
Run 116	**-22**	**18**	**-26**	**-30**
Run 117	**-18**	**-22**	**-30**	**26**
Run 118	**-14**	**10**	**2**	**-6**
Run 119	**-10**	**-14**	**6**	**2**
Run 120	**-6**	**2**	**-10**	**14**
Run 121	**-2**	**-6**	**-14**	**-10**
Run 122	**30**	**-26**	**-18**	**-22**
Run 123	**26**	**30**	**-22**	**18**
Run 124	**22**	**-18**	**26**	**30**
Run 125	**18**	**22**	**30**	**-26**
Run 126	**14**	**-10**	**-2**	**6**
Run 127	**10**	**14**	**-6**	**-2**
Run 128	**6**	**-2**	**10**	**-14**
Run 129	**2**	**6**	**14**	**10**
Run 130	**-127**	**125**	**121**	**123**
Run 131	**-125**	**-127**	**123**	**-121**
Run 132	**-123**	**121**	**-125**	**-127**
Run 133	**-121**	**-123**	**-127**	**125**
Run 134	**-119**	**117**	**113**	**-115**
Run 135	**-117**	**-119**	**115**	**113**
Run 136	**-115**	**113**	**-117**	**119**
Run 137	**-113**	**-115**	**-119**	**-117**
Run 138	**127**	**-125**	**-121**	**-123**
Run 139	**125**	**127**	**-123**	**121**
Run 140	**123**	**-121**	**125**	**127**
Run 141	**121**	**123**	**127**	**-125**
Run 142	**119**	**-117**	**-113**	**115**
Run 143	**117**	**119**	**-115**	**-113**
Run 144	**115**	**-113**	**117**	**-119**
Run 145	**113**	**115**	**119**	**117**
Run 146	**-111**	**109**	**105**	**107**
Run 147	**-109**	**-111**	**107**	**-105**

Run 148	**-107**	**105**	**-109**
Run 149	**-105**	**-107**	**-111**
Run 150	**-103**	**101**	**97**
Run 151	**-101**	**-103**	**99**
Run 152	**-99**	**97**	**-101**
Run 153	**-97**	**-99**	**-103**
Run 154	**111**	**-109**	**-105**
Run 155	**109**	**111**	**-107**
Run 156	**107**	**-105**	**109**
Run 157	**105**	**107**	**111**
Run 158	**103**	**-101**	**-97**
Run 159	**101**	**103**	**-99**
Run 160	**99**	**-97**	**101**
Run 161	**97**	**99**	**103**
Run 162	**-95**	**93**	**89**
Run 163	**-93**	**-95**	**91**
Run 164	**-91**	**89**	**-93**
Run 165	**-89**	**-91**	**-95**
Run 166	**-87**	**85**	**81**
Run 167	**-85**	**-87**	**83**
Run 168	**-83**	**81**	**-85**
Run 169	**-81**	**-83**	**-87**
Run 170	**95**	**-93**	**-89**
Run 171	**93**	**95**	**-91**
Run 172	**91**	**-89**	**93**
Run 173	**89**	**91**	**95**
Run 174	**87**	**-85**	**-81**
Run 175	**85**	**87**	**-83**
Run 176	**83**	**-81**	**85**
Run 177	**81**	**83**	**87**
Run 178	**-79**	**77**	**73**
Run 179	**-77**	**-79**	**75**
Run 180	**-75**	**73**	**-77**
Run 181	**-73**	**-75**	**-79**
Run 182	**-71**	**69**	**65**
Run 183	**-69**	**-71**	**67**
Run 184	**-67**	**65**	**-69**
Run 185	**-65**	**-67**	**-71**

Run 148	**-107**	**105**	**-109**	**-111**
Run 149	**-105**	**-107**	**-111**	**109**
Run 150	**-103**	**101**	**97**	**-99**
Run 151	**-101**	**-103**	**99**	**97**
Run 152	**-99**	**97**	**-101**	**103**
Run 153	**-97**	**-99**	**-103**	**-101**
Run 154	**111**	**-109**	**-105**	**-107**
Run 155	**109**	**111**	**-107**	**105**
Run 156	**107**	**-105**	**109**	**111**
Run 157	**105**	**107**	**111**	**-109**
Run 158	**103**	**-101**	**-97**	**99**
Run 159	**101**	**103**	**-99**	**-97**
Run 160	**99**	**-97**	**101**	**-103**
Run 161	**97**	**99**	**103**	**101**
Run 162	**-95**	**93**	**89**	**91**
Run 163	**-93**	**-95**	**91**	**-89**
Run 164	**-91**	**89**	**-93**	**-95**
Run 165	**-89**	**-91**	**-95**	**93**
Run 166	**-87**	**85**	**81**	**-83**
Run 167	**-85**	**-87**	**83**	**81**
Run 168	**-83**	**81**	**-85**	**87**
Run 169	**-81**	**-83**	**-87**	**-85**
Run 170	**95**	**-93**	**-89**	**-91**
Run 171	**93**	**95**	**-91**	**89**
Run 172	**91**	**-89**	**93**	**95**
Run 173	**89**	**91**	**95**	**-93**
Run 174	**87**	**-85**	**-81**	**83**
Run 175	**85**	**87**	**-83**	**-81**
Run 176	**83**	**-81**	**85**	**-87**
Run 177	**81**	**83**	**87**	**85**
Run 178	**-79**	**77**	**73**	**75**
Run 179	**-77**	**-79**	**75**	**-73**
Run 180	**-75**	**73**	**-77**	**-79**
Run 181	**-73**	**-75**	**-79**	**77**
Run 182	**-71**	**69**	**65**	**-67**
Run 183	**-69**	**-71**	**67**	**65**
Run 184	**-67**	**65**	**-69**	**71**
Run 185	**-65**	**-67**	**-71**	**-69**

Run 186	**79**	**-77**	**-73**
Run 187	**77**	**79**	**-75**
Run 188	**75**	**-73**	**77**
Run 189	**73**	**75**	**79**
Run 190	**71**	**-69**	**-65**
Run 191	**69**	**71**	**-67**
Run 192	**67**	**-65**	**69**
Run 193	**65**	**67**	**71**
Run 194	**-63**	**61**	**57**
Run 195	**-61**	**-63**	**59**
Run 196	**-59**	**57**	**-61**
Run 197	**-57**	**-59**	**-63**
Run 198	**-55**	**53**	**49**
Run 199	**-53**	**-55**	**51**
Run 200	**-51**	**49**	**-53**
Run 201	**-49**	**-51**	**-55**
Run 202	**63**	**-61**	**-57**
Run 203	**61**	**63**	**-59**
Run 204	**59**	**-57**	**61**
Run 205	**57**	**59**	**63**
Run 206	**55**	**-53**	**-49**
Run 207	**53**	**55**	**-51**
Run 208	**51**	**-49**	**53**
Run 209	**49**	**51**	**55**
Run 210	**-47**	**45**	**41**
Run 211	**-45**	**-47**	**43**
Run 212	**-43**	**41**	**-45**
Run 213	**-41**	**-43**	**-47**
Run 214	**-39**	**37**	**33**
Run 215	**-37**	**-39**	**35**
Run 216	**-35**	**33**	**-37**
Run 217	**-33**	**-35**	**-39**
Run 218	**47**	**-45**	**-41**
Run 219	**45**	**47**	**-43**
Run 220	**43**	**-41**	**45**
Run 221	**41**	**43**	**47**
Run 222	**39**	**-37**	**-33**
Run 223	**37**	**39**	**-35**

Run 186	**79**	**-77**	**-73**	**-75**
Run 187	**77**	**79**	**-75**	**73**
Run 188	**75**	**-73**	**77**	**79**
Run 189	**73**	**75**	**79**	**-77**
Run 190	**71**	**-69**	**-65**	**67**
Run 191	**69**	**71**	**-67**	**-65**
Run 192	**67**	**-65**	**69**	**-71**
Run 193	**65**	**67**	**71**	**69**
Run 194	**-63**	**61**	**57**	**59**
Run 195	**-61**	**-63**	**59**	**-57**
Run 196	**-59**	**57**	**-61**	**-63**
Run 197	**-57**	**-59**	**-63**	**61**
Run 198	**-55**	**53**	**49**	**-51**
Run 199	**-53**	**-55**	**51**	**49**
Run 200	**-51**	**49**	**-53**	**55**
Run 201	**-49**	**-51**	**-55**	**-53**
Run 202	**63**	**-61**	**-57**	**-59**
Run 203	**61**	**63**	**-59**	**57**
Run 204	**59**	**-57**	**61**	**63**
Run 205	**57**	**59**	**63**	**-61**
Run 206	**55**	**-53**	**-49**	**51**
Run 207	**53**	**55**	**-51**	**-49**
Run 208	**51**	**-49**	**53**	**-55**
Run 209	**49**	**51**	**55**	**53**
Run 210	**-47**	**45**	**41**	**43**
Run 211	**-45**	**-47**	**43**	**-41**
Run 212	**-43**	**41**	**-45**	**-47**
Run 213	**-41**	**-43**	**-47**	**45**
Run 214	**-39**	**37**	**33**	**-35**
Run 215	**-37**	**-39**	**35**	**33**
Run 216	**-35**	**33**	**-37**	**39**
Run 217	**-33**	**-35**	**-39**	**-37**
Run 218	**47**	**-45**	**-41**	**-43**
Run 219	**45**	**47**	**-43**	**41**
Run 220	**43**	**-41**	**45**	**47**
Run 221	**41**	**43**	**47**	**-45**
Run 222	**39**	**-37**	**-33**	**35**
Run 223	**37**	**39**	**-35**	**-33**

Run 224	**35**	**-33**	**37**
Run 225	**33**	**35**	**39**
Run 226	**-31**	**29**	**25**
Run 227	**-29**	**-31**	**27**
Run 228	**-27**	**25**	**-29**
Run 229	**-25**	**-27**	**-31**
Run 230	**-23**	**21**	**17**
Run 231	**-21**	**-23**	**19**
Run 232	**-19**	**17**	**-21**
Run 233	**-17**	**-19**	**-23**
Run 234	**31**	**-29**	**-25**
Run 235	**29**	**31**	**-27**
Run 236	**27**	**-25**	**29**
Run 237	**25**	**27**	**31**
Run 238	**23**	**-21**	**-17**
Run 239	**21**	**23**	**-19**
Run 240	**19**	**-17**	**21**
Run 241	**17**	**19**	**23**
Run 242	**-15**	**13**	**9**
Run 243	**-13**	**-15**	**11**
Run 244	**-11**	**9**	**-13**
Run 245	**-9**	**-11**	**-15**
Run 246	**-7**	**5**	**1**
Run 247	**-5**	**-7**	**3**
Run 248	**-3**	**1**	**-5**
Run 249	**-1**	**-3**	**-7**
Run 250	**15**	**-13**	**-9**
Run 251	**13**	**15**	**-11**
Run 252	**11**	**-9**	**13**
Run 253	**9**	**11**	**15**
Run 254	**7**	**-5**	**-1**
Run 255	**5**	**7**	**-3**
Run 256	**3**	**-1**	**5**
Run 257	**1**	**3**	**7**

Layer1- Run 1 to Run 257
Layer2- Run 1 to Run 129
Layer3- Run 1 to Run 65
Layer4- Run 1 to Run 33

Run 224	**35**	**-33**	**37**	**-39**
Run 225	**33**	**35**	**39**	**37**
Run 226	**-31**	**29**	**25**	**27**
Run 227	**-29**	**-31**	**27**	**-25**
Run 228	**-27**	**25**	**-29**	**-31**
Run 229	**-25**	**-27**	**-31**	**29**
Run 230	**-23**	**21**	**17**	**-19**
Run 231	**-21**	**-23**	**19**	**17**
Run 232	**-19**	**17**	**-21**	**23**
Run 233	**-17**	**-19**	**-23**	**-21**
Run 234	**31**	**-29**	**-25**	**-27**
Run 235	**29**	**31**	**-27**	**25**
Run 236	**27**	**-25**	**29**	**31**
Run 237	**25**	**27**	**31**	**-29**
Run 238	**23**	**-21**	**-17**	**19**
Run 239	**21**	**23**	**-19**	**-17**
Run 240	**19**	**-17**	**21**	**-23**
Run 241	**17**	**19**	**23**	**21**
Run 242	**-15**	**13**	**9**	**11**
Run 243	**-13**	**-15**	**11**	**-9**
Run 244	**-11**	**9**	**-13**	**-15**
Run 245	**-9**	**-11**	**-15**	**13**
Run 246	**-7**	**5**	**1**	**-3**
Run 247	**-5**	**-7**	**3**	**1**
Run 248	**-3**	**1**	**-5**	**7**
Run 249	**-1**	**-3**	**-7**	**-5**
Run 250	**15**	**-13**	**-9**	**-11**
Run 251	**13**	**15**	**-11**	**9**
Run 252	**11**	**-9**	**13**	**15**
Run 253	**9**	**11**	**15**	**-13**
Run 254	**7**	**-5**	**-1**	**3**
Run 255	**5**	**7**	**-3**	**-1**
Run 256	**3**	**-1**	**5**	**-7**
Run 257	**1**	**3**	**7**	**5**

Layer1- Run 1 to Run 257
Layer2- Run 1 to Run 129
Layer3- Run 1 to Run 65
Layer4- Run 1 to Run 33

	Layer5- Run 1 to Run 17 **Layer6-** Run 1 to Run 9	**Layer5-** Run 1 to Run 17 **Layer6-** Run 1 to Run 9

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Run 1	-176	-176	-48	-176	-112
Run 2	-144	-80	-80	176	144
Run 3	-112	144	176	-144	-16
Run 4	-80	16	16	16	16
Run 5	-48	80	-16	48	176
Run 6	-16	176	80	112	-80
Run 7	16	48	-176	80	-176
Run 8	48	-48	48	-48	48
Run 9	80	-144	112	144	-144
Run 10	112	-16	-144	-112	112
Run 11	144	112	-112	-80	-48
Run 12	176	-112	144	-16	80
Run 13	-192	-192	-192	0	-160
Run 14	-160	32	128	-32	192
Run 15	-128	192	-128	160	160
Run 16	-96	64	192	-128	-192
Run 17	-64	-64	64	64	-64
Run 18	-32	96	32	32	-96
Run 19	0	128	-64	-192	32
Run 20	32	-128	-160	-64	96
Run 21	64	-160	160	192	64
Run 22	96	-96	96	-160	128
Run 23	128	-32	0	128	-32
Run 24	160	160	-32	96	-128
Run 25	192	0	-96	-96	0
Run 26	120	-40	152	184	-168
Run 27	152	120	-40	-168	136

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5	Factor 6
Run 1	-176	-176	-48	-176	-112	-112
Run 2	-144	-80	-80	176	144	16
Run 3	-112	144	176	-144	-16	48
Run 4	-80	16	16	16	16	176
Run 5	-48	80	-16	48	176	-144
Run 6	-16	176	80	112	-80	-48
Run 7	16	48	-176	80	-176	-80
Run 8	48	-48	48	-48	48	80
Run 9	80	-144	112	144	-144	112
Run 10	112	-16	-144	-112	112	144
Run 11	144	112	-112	-80	-48	-16
Run 12	176	-112	144	-16	80	-176
Run 13	-192	-192	-192	0	-160	-32
Run 14	-160	32	128	-32	192	160
Run 15	-128	192	-128	160	160	-64
Run 16	-96	64	192	-128	-192	64
Run 17	-64	-64	64	64	-64	-128
Run 18	-32	96	32	32	-96	96
Run 19	0	128	-64	-192	32	-160
Run 20	32	-128	-160	-64	96	128
Run 21	64	-160	160	192	64	32
Run 22	96	-96	96	-160	128	-192
Run 23	128	-32	0	128	-32	-96
Run 24	160	160	-32	96	-128	0
Run 25	192	0	-96	-96	0	192
Run 26	120	-40	152	184	-168	136
Run 27	152	120	-40	-168	136	184

Run 28	**-40**	**152**	**120**	**136**	**184**
Run 29	**-184**	**168**	**-136**	**120**	**-40**
Run 30	**168**	**-136**	**-184**	**152**	**120**
Run 31	**-136**	**-184**	**168**	**-40**	**152**
Run 32	**56**	**-24**	**-8**	**88**	**104**
Run 33	**-24**	**-8**	**56**	**104**	**72**
Run 34	**-8**	**56**	**-24**	**72**	**88**
Run 35	**-104**	**-88**	**-72**	**-24**	**56**
Run 36	**-88**	**-72**	**-104**	**56**	**-8**
Run 37	**-72**	**-104**	**-88**	**-8**	**-24**
Run 38	**-120**	**40**	**-152**	**-184**	**168**
Run 39	**-152**	**-120**	**40**	**168**	**-136**
Run 40	**40**	**-152**	**-120**	**-136**	**-184**
Run 41	**184**	**-168**	**136**	**-120**	**40**
Run 42	**-168**	**136**	**184**	**-152**	**-120**
Run 43	**136**	**184**	**-168**	**40**	**-152**
Run 44	**-56**	**24**	**8**	**-88**	**-104**
Run 45	**24**	**8**	**-56**	**-104**	**-72**
Run 46	**8**	**-56**	**24**	**-72**	**-88**
Run 47	**104**	**88**	**72**	**24**	**-56**
Run 48	**88**	**72**	**104**	**-56**	**8**
Run 49	**72**	**104**	**88**	**8**	**24**
Run 50	**-188**	**180**	**164**	**172**	**132**
Run 51	**-180**	**-188**	**172**	**-164**	**140**
Run 52	**-172**	**164**	**-180**	**-188**	**148**
Run 53	**-164**	**-172**	**-188**	**180**	**156**
Run 54	**-156**	**148**	**132**	**-140**	**-164**
Run 55	**-148**	**-156**	**140**	**132**	**-172**
Run 56	**-140**	**132**	**-148**	**156**	**-180**
Run 57	**-132**	**-140**	**-156**	**-148**	**-188**
Run 58	**188**	**-180**	**-164**	**-172**	**-132**
Run 59	**180**	**188**	**-172**	**164**	**-140**
Run 60	**172**	**-164**	**180**	**188**	**-148**
Run 61	**164**	**172**	**188**	**-180**	**-156**
Run 62	**156**	**-148**	**-132**	**140**	**164**
Run 63	**148**	**156**	**-140**	**-132**	**172**
Run 64	**140**	**-132**	**148**	**-156**	**180**
Run 65	**132**	**140**	**156**	**148**	**188**

Run 28	**-40**	**152**	**120**	**136**	**184**	**-168**
Run 29	**-184**	**168**	**-136**	**120**	**-40**	**152**
Run 30	**168**	**-136**	**-184**	**152**	**120**	**-40**
Run 31	**-136**	**-184**	**168**	**-40**	**152**	**120**
Run 32	**56**	**-24**	**-8**	**88**	**104**	**72**
Run 33	**-24**	**-8**	**56**	**104**	**72**	**88**
Run 34	**-8**	**56**	**-24**	**72**	**88**	**104**
Run 35	**-104**	**-88**	**-72**	**-24**	**56**	**-8**
Run 36	**-88**	**-72**	**-104**	**56**	**-8**	**-24**
Run 37	**-72**	**-104**	**-88**	**-8**	**-24**	**56**
Run 38	**-120**	**40**	**-152**	**-184**	**168**	**-136**
Run 39	**-152**	**-120**	**40**	**168**	**-136**	**-184**
Run 40	**40**	**-152**	**-120**	**-136**	**-184**	**168**
Run 41	**184**	**-168**	**136**	**-120**	**40**	**-152**
Run 42	**-168**	**136**	**184**	**-152**	**-120**	**40**
Run 43	**136**	**184**	**-168**	**40**	**-152**	**-120**
Run 44	**-56**	**24**	**8**	**-88**	**-104**	**-72**
Run 45	**24**	**8**	**-56**	**-104**	**-72**	**-88**
Run 46	**8**	**-56**	**24**	**-72**	**-88**	**-104**
Run 47	**104**	**88**	**72**	**24**	**-56**	**8**
Run 48	**88**	**72**	**104**	**-56**	**8**	**24**
Run 49	**72**	**104**	**88**	**8**	**24**	**-56**
Run 50	**-188**	**180**	**164**	**172**	**132**	**-156**
Run 51	**-180**	**-188**	**172**	**-164**	**140**	**148**
Run 52	**-172**	**164**	**-180**	**-188**	**148**	**-140**
Run 53	**-164**	**-172**	**-188**	**180**	**156**	**132**
Run 54	**-156**	**148**	**132**	**-140**	**-164**	**188**
Run 55	**-148**	**-156**	**140**	**132**	**-172**	**-180**
Run 56	**-140**	**132**	**-148**	**156**	**-180**	**172**
Run 57	**-132**	**-140**	**-156**	**-148**	**-188**	**-164**
Run 58	**188**	**-180**	**-164**	**-172**	**-132**	**156**
Run 59	**180**	**188**	**-172**	**164**	**-140**	**-148**
Run 60	**172**	**-164**	**180**	**188**	**-148**	**140**
Run 61	**164**	**172**	**188**	**-180**	**-156**	**-132**
Run 62	**156**	**-148**	**-132**	**140**	**164**	**-188**
Run 63	**148**	**156**	**-140**	**-132**	**172**	**180**
Run 64	**140**	**-132**	**148**	**-156**	**180**	**-172**
Run 65	**132**	**140**	**156**	**148**	**188**	**164**

Run 66	**-124**	**116**	**100**	**108**	**68**
Run 67	**-116**	**-124**	**108**	**-100**	**76**
Run 68	**-108**	**100**	**-116**	**-124**	**84**
Run 69	**-100**	**-108**	**-124**	**116**	**92**
Run 70	**-92**	**84**	**68**	**-76**	**-100**
Run 71	**-84**	**-92**	**76**	**68**	**-108**
Run 72	**-76**	**68**	**-84**	**92**	**-116**
Run 73	**-68**	**-76**	**-92**	**-84**	**-124**
Run 74	**124**	**-116**	**-100**	**-108**	**-68**
Run 75	**116**	**124**	**-108**	**100**	**-76**
Run 76	**108**	**-100**	**116**	**124**	**-84**
Run 77	**100**	**108**	**124**	**-116**	**-92**
Run 78	**92**	**-84**	**-68**	**76**	**100**
Run 79	**84**	**92**	**-76**	**-68**	**108**
Run 80	**76**	**-68**	**84**	**-92**	**116**
Run 81	**68**	**76**	**92**	**84**	**124**
Run 82	**-60**	**52**	**36**	**44**	**4**
Run 83	**-52**	**-60**	**44**	**-36**	**12**
Run 84	**-44**	**36**	**-52**	**-60**	**20**
Run 85	**-36**	**-44**	**-60**	**52**	**28**
Run 86	**-28**	**20**	**4**	**-12**	**-36**
Run 87	**-20**	**-28**	**12**	**4**	**-44**
Run 88	**-12**	**4**	**-20**	**28**	**-52**
Run 89	**-4**	**-12**	**-28**	**-20**	**-60**
Run 90	**60**	**-52**	**-36**	**-44**	**-4**
Run 91	**52**	**60**	**-44**	**36**	**-12**
Run 92	**44**	**-36**	**52**	**60**	**-20**
Run 93	**36**	**44**	**60**	**-52**	**-28**
Run 94	**28**	**-20**	**-4**	**12**	**36**
Run 95	**20**	**28**	**-12**	**-4**	**44**
Run 96	**12**	**-4**	**20**	**-28**	**52**
Run 97	**4**	**12**	**28**	**20**	**60**
Run 98	**-190**	**186**	**178**	**182**	**162**
Run 99	**-186**	**-190**	**182**	**-178**	**166**
Run 100	**-182**	**178**	**-186**	**-190**	**170**
Run 101	**-178**	**-182**	**-190**	**186**	**174**
Run 102	**-174**	**170**	**162**	**-166**	**-178**
Run 103	**-170**	**-174**	**166**	**162**	**-182**

Run 66	**-124**	**116**	**100**	**108**	**68**	**-92**
Run 67	**-116**	**-124**	**108**	**-100**	**76**	**84**
Run 68	**-108**	**100**	**-116**	**-124**	**84**	**-76**
Run 69	**-100**	**-108**	**-124**	**116**	**92**	**68**
Run 70	**-92**	**84**	**68**	**-76**	**-100**	**124**
Run 71	**-84**	**-92**	**76**	**68**	**-108**	**-116**
Run 72	**-76**	**68**	**-84**	**92**	**-116**	**108**
Run 73	**-68**	**-76**	**-92**	**-84**	**-124**	**-100**
Run 74	**124**	**-116**	**-100**	**-108**	**-68**	**92**
Run 75	**116**	**124**	**-108**	**100**	**-76**	**-84**
Run 76	**108**	**-100**	**116**	**124**	**-84**	**76**
Run 77	**100**	**108**	**124**	**-116**	**-92**	**-68**
Run 78	**92**	**-84**	**-68**	**76**	**100**	**-124**
Run 79	**84**	**92**	**-76**	**-68**	**108**	**116**
Run 80	**76**	**-68**	**84**	**-92**	**116**	**-108**
Run 81	**68**	**76**	**92**	**84**	**124**	**100**
Run 82	**-60**	**52**	**36**	**44**	**4**	**-28**
Run 83	**-52**	**-60**	**44**	**-36**	**12**	**20**
Run 84	**-44**	**36**	**-52**	**-60**	**20**	**-12**
Run 85	**-36**	**-44**	**-60**	**52**	**28**	**4**
Run 86	**-28**	**20**	**4**	**-12**	**-36**	**60**
Run 87	**-20**	**-28**	**12**	**4**	**-44**	**-52**
Run 88	**-12**	**4**	**-20**	**28**	**-52**	**44**
Run 89	**-4**	**-12**	**-28**	**-20**	**-60**	**-36**
Run 90	**60**	**-52**	**-36**	**-44**	**-4**	**28**
Run 91	**52**	**60**	**-44**	**36**	**-12**	**-20**
Run 92	**44**	**-36**	**52**	**60**	**-20**	**12**
Run 93	**36**	**44**	**60**	**-52**	**-28**	**-4**
Run 94	**28**	**-20**	**-4**	**12**	**36**	**-60**
Run 95	**20**	**28**	**-12**	**-4**	**44**	**52**
Run 96	**12**	**-4**	**20**	**-28**	**52**	**-44**
Run 97	**4**	**12**	**28**	**20**	**60**	**36**
Run 98	**-190**	**186**	**178**	**182**	**162**	**-174**
Run 99	**-186**	**-190**	**182**	**-178**	**166**	**170**
Run 100	**-182**	**178**	**-186**	**-190**	**170**	**-166**
Run 101	**-178**	**-182**	**-190**	**186**	**174**	**162**
Run 102	**-174**	**170**	**162**	**-166**	**-178**	**190**
Run 103	**-170**	**-174**	**166**	**162**	**-182**	**-186**

Run 104	**-166**	**162**	**-170**	**174**	**-186**
Run 105	**-162**	**-166**	**-174**	**-170**	**-190**
Run 106	**190**	**-186**	**-178**	**-182**	**-162**
Run 107	**186**	**190**	**-182**	**178**	**-166**
Run 108	**182**	**-178**	**186**	**190**	**-170**
Run 109	**178**	**182**	**190**	**-186**	**-174**
Run 110	**174**	**-170**	**-162**	**166**	**178**
Run 111	**170**	**174**	**-166**	**-162**	**182**
Run 112	**166**	**-162**	**170**	**-174**	**186**
Run 113	**162**	**166**	**174**	**170**	**190**
Run 114	**-158**	**154**	**146**	**150**	**130**
Run 115	**-154**	**-158**	**150**	**-146**	**134**
Run 116	**-150**	**146**	**-154**	**-158**	**138**
Run 117	**-146**	**-150**	**-158**	**154**	**142**
Run 118	**-142**	**138**	**130**	**-134**	**-146**
Run 119	**-138**	**-142**	**134**	**130**	**-150**
Run 120	**-134**	**130**	**-138**	**142**	**-154**
Run 121	**-130**	**-134**	**-142**	**-138**	**-158**
Run 122	**158**	**-154**	**-146**	**-150**	**-130**
Run 123	**154**	**158**	**-150**	**146**	**-134**
Run 124	**150**	**-146**	**154**	**158**	**-138**
Run 125	**146**	**150**	**158**	**-154**	**-142**
Run 126	**142**	**-138**	**-130**	**134**	**146**
Run 127	**138**	**142**	**-134**	**-130**	**150**
Run 128	**134**	**-130**	**138**	**-142**	**154**
Run 129	**130**	**134**	**142**	**138**	**158**
Run 130	**-126**	**122**	**114**	**118**	**98**
Run 131	**-122**	**-126**	**118**	**-114**	**102**
Run 132	**-118**	**114**	**-122**	**-126**	**106**
Run 133	**-114**	**-118**	**-126**	**122**	**110**
Run 134	**-110**	**106**	**98**	**-102**	**-114**
Run 135	**-106**	**-110**	**102**	**98**	**-118**
Run 136	**-102**	**98**	**-106**	**110**	**-122**
Run 137	**-98**	**-102**	**-110**	**-106**	**-126**
Run 138	**126**	**-122**	**-114**	**-118**	**-98**
Run 139	**122**	**126**	**-118**	**114**	**-102**
Run 140	**118**	**-114**	**122**	**126**	**-106**
Run 141	**114**	**118**	**126**	**-122**	**-110**

Run 104	**-166**	**162**	**-170**	**174**	**-186**	**182**
Run 105	**-162**	**-166**	**-174**	**-170**	**-190**	**-178**
Run 106	**190**	**-186**	**-178**	**-182**	**-162**	**174**
Run 107	**186**	**190**	**-182**	**178**	**-166**	**-170**
Run 108	**182**	**-178**	**186**	**190**	**-170**	**166**
Run 109	**178**	**182**	**190**	**-186**	**-174**	**-162**
Run 110	**174**	**-170**	**-162**	**166**	**178**	**-190**
Run 111	**170**	**174**	**-166**	**-162**	**182**	**186**
Run 112	**166**	**-162**	**170**	**-174**	**186**	**-182**
Run 113	**162**	**166**	**174**	**170**	**190**	**178**
Run 114	**-158**	**154**	**146**	**150**	**130**	**-142**
Run 115	**-154**	**-158**	**150**	**-146**	**134**	**138**
Run 116	**-150**	**146**	**-154**	**-158**	**138**	**-134**
Run 117	**-146**	**-150**	**-158**	**154**	**142**	**130**
Run 118	**-142**	**138**	**130**	**-134**	**-146**	**158**
Run 119	**-138**	**-142**	**134**	**130**	**-150**	**-154**
Run 120	**-134**	**130**	**-138**	**142**	**-154**	**150**
Run 121	**-130**	**-134**	**-142**	**-138**	**-158**	**-146**
Run 122	**158**	**-154**	**-146**	**-150**	**-130**	**142**
Run 123	**154**	**158**	**-150**	**146**	**-134**	**-138**
Run 124	**150**	**-146**	**154**	**158**	**-138**	**134**
Run 125	**146**	**150**	**158**	**-154**	**-142**	**-130**
Run 126	**142**	**-138**	**-130**	**134**	**146**	**-158**
Run 127	**138**	**142**	**-134**	**-130**	**150**	**154**
Run 128	**134**	**-130**	**138**	**-142**	**154**	**-150**
Run 129	**130**	**134**	**142**	**138**	**158**	**146**
Run 130	**-126**	**122**	**114**	**118**	**98**	**-110**
Run 131	**-122**	**-126**	**118**	**-114**	**102**	**106**
Run 132	**-118**	**114**	**-122**	**-126**	**106**	**-102**
Run 133	**-114**	**-118**	**-126**	**122**	**110**	**98**
Run 134	**-110**	**106**	**98**	**-102**	**-114**	**126**
Run 135	**-106**	**-110**	**102**	**98**	**-118**	**-122**
Run 136	**-102**	**98**	**-106**	**110**	**-122**	**118**
Run 137	**-98**	**-102**	**-110**	**-106**	**-126**	**-114**
Run 138	**126**	**-122**	**-114**	**-118**	**-98**	**110**
Run 139	**122**	**126**	**-118**	**114**	**-102**	**-106**
Run 140	**118**	**-114**	**122**	**126**	**-106**	**102**
Run 141	**114**	**118**	**126**	**-122**	**-110**	**-98**

Run 142	**110**	**-106**	**-98**	**102**	**114**
Run 143	**106**	**110**	**-102**	**-98**	**118**
Run 144	**102**	**-98**	**106**	**-110**	**122**
Run 145	**98**	**102**	**110**	**106**	**126**
Run 146	**-94**	**90**	**82**	**86**	**66**
Run 147	**-90**	**-94**	**86**	**-82**	**70**
Run 148	**-86**	**82**	**-90**	**-94**	**74**
Run 149	**-82**	**-86**	**-94**	**90**	**78**
Run 150	**-78**	**74**	**66**	**-70**	**-82**
Run 151	**-74**	**-78**	**70**	**66**	**-86**
Run 152	**-70**	**66**	**-74**	**78**	**-90**
Run 153	**-66**	**-70**	**-78**	**-74**	**-94**
Run 154	**94**	**-90**	**-82**	**-86**	**-66**
Run 155	**90**	**94**	**-86**	**82**	**-70**
Run 156	**86**	**-82**	**90**	**94**	**-74**
Run 157	**82**	**86**	**94**	**-90**	**-78**
Run 158	**78**	**-74**	**-66**	**70**	**82**
Run 159	**74**	**78**	**-70**	**-66**	**86**
Run 160	**70**	**-66**	**74**	**-78**	**90**
Run 161	**66**	**70**	**78**	**74**	**94**
Run 162	**-62**	**58**	**50**	**54**	**34**
Run 163	**-58**	**-62**	**54**	**-50**	**38**
Run 164	**-54**	**50**	**-58**	**-62**	**42**
Run 165	**-50**	**-54**	**-62**	**58**	**46**
Run 166	**-46**	**42**	**34**	**-38**	**-50**
Run 167	**-42**	**-46**	**38**	**34**	**-54**
Run 168	**-38**	**34**	**-42**	**46**	**-58**
Run 169	**-34**	**-38**	**-46**	**-42**	**-62**
Run 170	**62**	**-58**	**-50**	**-54**	**-34**
Run 171	**58**	**62**	**-54**	**50**	**-38**
Run 172	**54**	**-50**	**58**	**62**	**-42**
Run 173	**50**	**54**	**62**	**-58**	**-46**
Run 174	**46**	**-42**	**-34**	**38**	**50**
Run 175	**42**	**46**	**-38**	**-34**	**54**
Run 176	**38**	**-34**	**42**	**-46**	**58**
Run 177	**34**	**38**	**46**	**42**	**62**
Run 178	**-30**	**26**	**18**	**22**	**2**
Run 179	**-26**	**-30**	**22**	**-18**	**6**

Run 142	**110**	**-106**	**-98**	**102**	**114**	**-126**
Run 143	**106**	**110**	**-102**	**-98**	**118**	**122**
Run 144	**102**	**-98**	**106**	**-110**	**122**	**-118**
Run 145	**98**	**102**	**110**	**106**	**126**	**114**
Run 146	**-94**	**90**	**82**	**86**	**66**	**-78**
Run 147	**-90**	**-94**	**86**	**-82**	**70**	**74**
Run 148	**-86**	**82**	**-90**	**-94**	**74**	**-70**
Run 149	**-82**	**-86**	**-94**	**90**	**78**	**66**
Run 150	**-78**	**74**	**66**	**-70**	**-82**	**94**
Run 151	**-74**	**-78**	**70**	**66**	**-86**	**-90**
Run 152	**-70**	**66**	**-74**	**78**	**-90**	**86**
Run 153	**-66**	**-70**	**-78**	**-74**	**-94**	**-82**
Run 154	**94**	**-90**	**-82**	**-86**	**-66**	**78**
Run 155	**90**	**94**	**-86**	**82**	**-70**	**-74**
Run 156	**86**	**-82**	**90**	**94**	**-74**	**70**
Run 157	**82**	**86**	**94**	**-90**	**-78**	**-66**
Run 158	**78**	**-74**	**-66**	**70**	**82**	**-94**
Run 159	**74**	**78**	**-70**	**-66**	**86**	**90**
Run 160	**70**	**-66**	**74**	**-78**	**90**	**-86**
Run 161	**66**	**70**	**78**	**74**	**94**	**82**
Run 162	**-62**	**58**	**50**	**54**	**34**	**-46**
Run 163	**-58**	**-62**	**54**	**-50**	**38**	**42**
Run 164	**-54**	**50**	**-58**	**-62**	**42**	**-38**
Run 165	**-50**	**-54**	**-62**	**58**	**46**	**34**
Run 166	**-46**	**42**	**34**	**-38**	**-50**	**62**
Run 167	**-42**	**-46**	**38**	**34**	**-54**	**-58**
Run 168	**-38**	**34**	**-42**	**46**	**-58**	**54**
Run 169	**-34**	**-38**	**-46**	**-42**	**-62**	**-50**
Run 170	**62**	**-58**	**-50**	**-54**	**-34**	**46**
Run 171	**58**	**62**	**-54**	**50**	**-38**	**-42**
Run 172	**54**	**-50**	**58**	**62**	**-42**	**38**
Run 173	**50**	**54**	**62**	**-58**	**-46**	**-34**
Run 174	**46**	**-42**	**-34**	**38**	**50**	**-62**
Run 175	**42**	**46**	**-38**	**-34**	**54**	**58**
Run 176	**38**	**-34**	**42**	**-46**	**58**	**-54**
Run 177	**34**	**38**	**46**	**42**	**62**	**50**
Run 178	**-30**	**26**	**18**	**22**	**2**	**-14**
Run 179	**-26**	**-30**	**22**	**-18**	**6**	**10**

Run 180	-22	18	-26	-30	10
Run 181	-18	-22	-30	26	14
Run 182	-14	10	2	-6	-18
Run 183	-10	-14	6	2	-22
Run 184	-6	2	-10	14	-26
Run 185	-2	-6	-14	-10	-30
Run 186	30	-26	-18	-22	-2
Run 187	26	30	-22	18	-6
Run 188	22	-18	26	30	-10
Run 189	18	22	30	-26	-14
Run 190	14	-10	-2	6	18
Run 191	10	14	-6	-2	22
Run 192	6	-2	10	-14	26
Run 193	2	6	14	10	30
Run 194	-191	189	185	187	177
Run 195	-189	-191	187	-185	179
Run 196	-187	185	-189	-191	181
Run 197	-185	-187	-191	189	183
Run 198	-183	181	177	-179	-185
Run 199	-181	-183	179	177	-187
Run 200	-179	177	-181	183	-189
Run 201	-177	-179	-183	-181	-191
Run 202	191	-189	-185	-187	-177
Run 203	189	191	-187	185	-179
Run 204	187	-185	189	191	-181
Run 205	185	187	191	-189	-183
Run 206	183	-181	-177	179	185
Run 207	181	183	-179	-177	187
Run 208	179	-177	181	-183	189
Run 209	177	179	183	181	191
Run 210	-175	173	169	171	161
Run 211	-173	-175	171	-169	163
Run 212	-171	169	-173	-175	165
Run 213	-169	-171	-175	173	167
Run 214	-167	165	161	-163	-169
Run 215	-165	-167	163	161	-171
Run 216	-163	161	-165	167	-173
Run 217	-161	-163	-167	-165	-175

Run 180	-22	18	-26	-30	10	-6
Run 181	-18	-22	-30	26	14	2
Run 182	-14	10	2	-6	-18	30
Run 183	-10	-14	6	2	-22	-26
Run 184	-6	2	-10	14	-26	22
Run 185	-2	-6	-14	-10	-30	-18
Run 186	30	-26	-18	-22	-2	14
Run 187	26	30	-22	18	-6	-10
Run 188	22	-18	26	30	-10	6
Run 189	18	22	30	-26	-14	-2
Run 190	14	-10	-2	6	18	-30
Run 191	10	14	-6	-2	22	26
Run 192	6	-2	10	-14	26	-22
Run 193	2	6	14	10	30	18
Run 194	-191	189	185	187	177	-183
Run 195	-189	-191	187	-185	179	181
Run 196	-187	185	-189	-191	181	-179
Run 197	-185	-187	-191	189	183	177
Run 198	-183	181	177	-179	-185	191
Run 199	-181	-183	179	177	-187	-189
Run 200	-179	177	-181	183	-189	187
Run 201	-177	-179	-183	-181	-191	-185
Run 202	191	-189	-185	-187	-177	183
Run 203	189	191	-187	185	-179	-181
Run 204	187	-185	189	191	-181	179
Run 205	185	187	191	-189	-183	-177
Run 206	183	-181	-177	179	185	-191
Run 207	181	183	-179	-177	187	189
Run 208	179	-177	181	-183	189	-187
Run 209	177	179	183	181	191	185
Run 210	-175	173	169	171	161	-167
Run 211	-173	-175	171	-169	163	165
Run 212	-171	169	-173	-175	165	-163
Run 213	-169	-171	-175	173	167	161
Run 214	-167	165	161	-163	-169	175
Run 215	-165	-167	163	161	-171	-173
Run 216	-163	161	-165	167	-173	171
Run 217	-161	-163	-167	-165	-175	-169

Run 218	175	-173	-169	-171	-161
Run 219	173	175	-171	169	-163
Run 220	171	-169	173	175	-165
Run 221	169	171	175	-173	-167
Run 222	167	-165	-161	163	169
Run 223	165	167	-163	-161	171
Run 224	163	-161	165	-167	173
Run 225	161	163	167	165	175
Run 226	-159	157	153	155	145
Run 227	-157	-159	155	-153	147
Run 228	-155	153	-157	-159	149
Run 229	-153	-155	-159	157	151
Run 230	-151	149	145	-147	-153
Run 231	-149	-151	147	145	-155
Run 232	-147	145	-149	151	-157
Run 233	-145	-147	-151	-149	-159
Run 234	159	-157	-153	-155	-145
Run 235	157	159	-155	153	-147
Run 236	155	-153	157	159	-149
Run 237	153	155	159	-157	-151
Run 238	151	-149	-145	147	153
Run 239	149	151	-147	-145	155
Run 240	147	-145	149	-151	157
Run 241	145	147	151	149	159
Run 242	-143	141	137	139	129
Run 243	-141	-143	139	-137	131
Run 244	-139	137	-141	-143	133
Run 245	-137	-139	-143	141	135
Run 246	-135	133	129	-131	-137
Run 247	-133	-135	131	129	-139
Run 248	-131	129	-133	135	-141
Run 249	-129	-131	-135	-133	-143
Run 250	143	-141	-137	-139	-129
Run 251	141	143	-139	137	-131
Run 252	139	-137	141	143	-133
Run 253	137	139	143	-141	-135
Run 254	135	-133	-129	131	137
Run 255	133	135	-131	-129	139

Run 218	175	-173	-169	-171	-161	167
Run 219	173	175	-171	169	-163	-165
Run 220	171	-169	173	175	-165	163
Run 221	169	171	175	-173	-167	-161
Run 222	167	-165	-161	163	169	-175
Run 223	165	167	-163	-161	171	173
Run 224	163	-161	165	-167	173	-171
Run 225	161	163	167	165	175	169
Run 226	-159	157	153	155	145	-151
Run 227	-157	-159	155	-153	147	149
Run 228	-155	153	-157	-159	149	-147
Run 229	-153	-155	-159	157	151	145
Run 230	-151	149	145	-147	-153	159
Run 231	-149	-151	147	145	-155	-157
Run 232	-147	145	-149	151	-157	155
Run 233	-145	-147	-151	-149	-159	-153
Run 234	159	-157	-153	-155	-145	151
Run 235	157	159	-155	153	-147	-149
Run 236	155	-153	157	159	-149	147
Run 237	153	155	159	-157	-151	-145
Run 238	151	-149	-145	147	153	-159
Run 239	149	151	-147	-145	155	157
Run 240	147	-145	149	-151	157	-155
Run 241	145	147	151	149	159	153
Run 242	-143	141	137	139	129	-135
Run 243	-141	-143	139	-137	131	133
Run 244	-139	137	-141	-143	133	-131
Run 245	-137	-139	-143	141	135	129
Run 246	-135	133	129	-131	-137	143
Run 247	-133	-135	131	129	-139	-141
Run 248	-131	129	-133	135	-141	139
Run 249	-129	-131	-135	-133	-143	-137
Run 250	143	-141	-137	-139	-129	135
Run 251	141	143	-139	137	-131	-133
Run 252	139	-137	141	143	-133	131
Run 253	137	139	143	-141	-135	-129
Run 254	135	-133	-129	131	137	-143
Run 255	133	135	-131	-129	139	141

Run 256	**131**	**-129**	**133**	**-135**	**141**
Run 257	**129**	**131**	**135**	**133**	**143**
Run 258	**-127**	**125**	**121**	**123**	**113**
Run 259	**-125**	**-127**	**123**	**-121**	**115**
Run 260	**-123**	**121**	**-125**	**-127**	**117**
Run 261	**-121**	**-123**	**-127**	**125**	**119**
Run 262	**-119**	**117**	**113**	**-115**	**-121**
Run 263	**-117**	**-119**	**115**	**113**	**-123**
Run 264	**-115**	**113**	**-117**	**119**	**-125**
Run 265	**-113**	**-115**	**-119**	**-117**	**-127**
Run 266	**127**	**-125**	**-121**	**-123**	**-113**
Run 267	**125**	**127**	**-123**	**121**	**-115**
Run 268	**123**	**-121**	**125**	**127**	**-117**
Run 269	**121**	**123**	**127**	**-125**	**-119**
Run 270	**119**	**-117**	**-113**	**115**	**121**
Run 271	**117**	**119**	**-115**	**-113**	**123**
Run 272	**115**	**-113**	**117**	**-119**	**125**
Run 273	**113**	**115**	**119**	**117**	**127**
Run 274	**-111**	**109**	**105**	**107**	**97**
Run 275	**-109**	**-111**	**107**	**-105**	**99**
Run 276	**-107**	**105**	**-109**	**-111**	**101**
Run 277	**-105**	**-107**	**-111**	**109**	**103**
Run 278	**-103**	**101**	**97**	**-99**	**-105**
Run 279	**-101**	**-103**	**99**	**97**	**-107**
Run 280	**-99**	**97**	**-101**	**103**	**-109**
Run 281	**-97**	**-99**	**-103**	**-101**	**-111**
Run 282	**111**	**-109**	**-105**	**-107**	**-97**
Run 283	**109**	**111**	**-107**	**105**	**-99**
Run 284	**107**	**-105**	**109**	**111**	**-101**
Run 285	**105**	**107**	**111**	**-109**	**-103**
Run 286	**103**	**-101**	**-97**	**99**	**105**
Run 287	**101**	**103**	**-99**	**-97**	**107**
Run 288	**99**	**-97**	**101**	**-103**	**109**
Run 289	**97**	**99**	**103**	**101**	**111**
Run 290	**-95**	**93**	**89**	**91**	**81**
Run 291	**-93**	**-95**	**91**	**-89**	**83**
Run 292	**-91**	**89**	**-93**	**-95**	**85**
Run 293	**-89**	**-91**	**-95**	**93**	**87**

Run 256	**131**	**-129**	**133**	**-135**	**141**	**-139**
Run 257	**129**	**131**	**135**	**133**	**143**	**137**
Run 258	**-127**	**125**	**121**	**123**	**113**	**-119**
Run 259	**-125**	**-127**	**123**	**-121**	**115**	**117**
Run 260	**-123**	**121**	**-125**	**-127**	**117**	**-115**
Run 261	**-121**	**-123**	**-127**	**125**	**119**	**113**
Run 262	**-119**	**117**	**113**	**-115**	**-121**	**127**
Run 263	**-117**	**-119**	**115**	**113**	**-123**	**-125**
Run 264	**-115**	**113**	**-117**	**119**	**-125**	**123**
Run 265	**-113**	**-115**	**-119**	**-117**	**-127**	**-121**
Run 266	**127**	**-125**	**-121**	**-123**	**-113**	**119**
Run 267	**125**	**127**	**-123**	**121**	**-115**	**-117**
Run 268	**123**	**-121**	**125**	**127**	**-117**	**115**
Run 269	**121**	**123**	**127**	**-125**	**-119**	**-113**
Run 270	**119**	**-117**	**-113**	**115**	**121**	**-127**
Run 271	**117**	**119**	**-115**	**-113**	**123**	**125**
Run 272	**115**	**-113**	**117**	**-119**	**125**	**-123**
Run 273	**113**	**115**	**119**	**117**	**127**	**121**
Run 274	**-111**	**109**	**105**	**107**	**97**	**-103**
Run 275	**-109**	**-111**	**107**	**-105**	**99**	**101**
Run 276	**-107**	**105**	**-109**	**-111**	**101**	**-99**
Run 277	**-105**	**-107**	**-111**	**109**	**103**	**97**
Run 278	**-103**	**101**	**97**	**-99**	**-105**	**111**
Run 279	**-101**	**-103**	**99**	**97**	**-107**	**-109**
Run 280	**-99**	**97**	**-101**	**103**	**-109**	**107**
Run 281	**-97**	**-99**	**-103**	**-101**	**-111**	**-105**
Run 282	**111**	**-109**	**-105**	**-107**	**-97**	**103**
Run 283	**109**	**111**	**-107**	**105**	**-99**	**-101**
Run 284	**107**	**-105**	**109**	**111**	**-101**	**99**
Run 285	**105**	**107**	**111**	**-109**	**-103**	**-97**
Run 286	**103**	**-101**	**-97**	**99**	**105**	**-111**
Run 287	**101**	**103**	**-99**	**-97**	**107**	**109**
Run 288	**99**	**-97**	**101**	**-103**	**109**	**-107**
Run 289	**97**	**99**	**103**	**101**	**111**	**105**
Run 290	**-95**	**93**	**89**	**91**	**81**	**-87**
Run 291	**-93**	**-95**	**91**	**-89**	**83**	**85**
Run 292	**-91**	**89**	**-93**	**-95**	**85**	**-83**
Run 293	**-89**	**-91**	**-95**	**93**	**87**	**81**

Run 294	-87	85	81	-83	-89
Run 295	-85	-87	83	81	-91
Run 296	-83	81	-85	87	-93
Run 297	-81	-83	-87	-85	-95
Run 298	95	-93	-89	-91	-81
Run 299	93	95	-91	89	-83
Run 300	91	-89	93	95	-85
Run 301	89	91	95	-93	-87
Run 302	87	-85	-81	83	89
Run 303	85	87	-83	-81	91
Run 304	83	-81	85	-87	93
Run 305	81	83	87	85	95
Run 306	-79	77	73	75	65
Run 307	-77	-79	75	-73	67
Run 308	-75	73	-77	-79	69
Run 309	-73	-75	-79	77	71
Run 310	-71	69	65	-67	-73
Run 311	-69	-71	67	65	-75
Run 312	-67	65	-69	71	-77
Run 313	-65	-67	-71	-69	-79
Run 314	79	-77	-73	-75	-65
Run 315	77	79	-75	73	-67
Run 316	75	-73	77	79	-69
Run 317	73	75	79	-77	-71
Run 318	71	-69	-65	67	73
Run 319	69	71	-67	-65	75
Run 320	67	-65	69	-71	77
Run 321	65	67	71	69	79
Run 322	-63	61	57	59	49
Run 323	-61	-63	59	-57	51
Run 324	-59	57	-61	-63	53
Run 325	-57	-59	-63	61	55
Run 326	-55	53	49	-51	-57
Run 327	-53	-55	51	49	-59
Run 328	-51	49	-53	55	-61
Run 329	-49	-51	-55	-53	-63
Run 330	63	-61	-57	-59	-49
Run 331	61	63	-59	57	-51

Run 294	-87	85	81	-83	-89	95
Run 295	-85	-87	83	81	-91	-93
Run 296	-83	81	-85	87	-93	91
Run 297	-81	-83	-87	-85	-95	-89
Run 298	95	-93	-89	-91	-81	87
Run 299	93	95	-91	89	-83	-85
Run 300	91	-89	93	95	-85	83
Run 301	89	91	95	-93	-87	-81
Run 302	87	-85	-81	83	89	-95
Run 303	85	87	-83	-81	91	93
Run 304	83	-81	85	-87	93	-91
Run 305	81	83	87	85	95	89
Run 306	-79	77	73	75	65	-71
Run 307	-77	-79	75	-73	67	69
Run 308	-75	73	-77	-79	69	-67
Run 309	-73	-75	-79	77	71	65
Run 310	-71	69	65	-67	-73	79
Run 311	-69	-71	67	65	-75	-77
Run 312	-67	65	-69	71	-77	75
Run 313	-65	-67	-71	-69	-79	-73
Run 314	79	-77	-73	-75	-65	71
Run 315	77	79	-75	73	-67	-69
Run 316	75	-73	77	79	-69	67
Run 317	73	75	79	-77	-71	-65
Run 318	71	-69	-65	67	73	-79
Run 319	69	71	-67	-65	75	77
Run 320	67	-65	69	-71	77	-75
Run 321	65	67	71	69	79	73
Run 322	-63	61	57	59	49	-55
Run 323	-61	-63	59	-57	51	53
Run 324	-59	57	-61	-63	53	-51
Run 325	-57	-59	-63	61	55	49
Run 326	-55	53	49	-51	-57	63
Run 327	-53	-55	51	49	-59	-61
Run 328	-51	49	-53	55	-61	59
Run 329	-49	-51	-55	-53	-63	-57
Run 330	63	-61	-57	-59	-49	55
Run 331	61	63	-59	57	-51	-53

Run 332	**59**	**-57**	**61**	**63**	**-53**
Run 333	**57**	**59**	**63**	**-61**	**-55**
Run 334	**55**	**-53**	**-49**	**51**	**57**
Run 335	**53**	**55**	**-51**	**-49**	**59**
Run 336	**51**	**-49**	**53**	**-55**	**61**
Run 337	**49**	**51**	**55**	**53**	**63**
Run 338	**-47**	**45**	**41**	**43**	**33**
Run 339	**-45**	**-47**	**43**	**-41**	**35**
Run 340	**-43**	**41**	**-45**	**-47**	**37**
Run 341	**-41**	**-43**	**-47**	**45**	**39**
Run 342	**-39**	**37**	**33**	**-35**	**-41**
Run 343	**-37**	**-39**	**35**	**33**	**-43**
Run 344	**-35**	**33**	**-37**	**39**	**-45**
Run 345	**-33**	**-35**	**-39**	**-37**	**-47**
Run 346	**47**	**-45**	**-41**	**-43**	**-33**
Run 347	**45**	**47**	**-43**	**41**	**-35**
Run 348	**43**	**-41**	**45**	**47**	**-37**
Run 349	**41**	**43**	**47**	**-45**	**-39**
Run 350	**39**	**-37**	**-33**	**35**	**41**
Run 351	**37**	**39**	**-35**	**-33**	**43**
Run 352	**35**	**-33**	**37**	**-39**	**45**
Run 353	**33**	**35**	**39**	**37**	**47**
Run 354	**-31**	**29**	**25**	**27**	**17**
Run 355	**-29**	**-31**	**27**	**-25**	**19**
Run 356	**-27**	**25**	**-29**	**-31**	**21**
Run 357	**-25**	**-27**	**-31**	**29**	**23**
Run 358	**-23**	**21**	**17**	**-19**	**-25**
Run 359	**-21**	**-23**	**19**	**17**	**-27**
Run 360	**-19**	**17**	**-21**	**23**	**-29**
Run 361	**-17**	**-19**	**-23**	**-21**	**-31**
Run 362	**31**	**-29**	**-25**	**-27**	**-17**
Run 363	**29**	**31**	**-27**	**25**	**-19**
Run 364	**27**	**-25**	**29**	**31**	**-21**
Run 365	**25**	**27**	**31**	**-29**	**-23**
Run 366	**23**	**-21**	**-17**	**19**	**25**
Run 367	**21**	**23**	**-19**	**-17**	**27**
Run 368	**19**	**-17**	**21**	**-23**	**29**
Run 369	**17**	**19**	**23**	**21**	**31**

Run 332	**59**	**-57**	**61**	**63**	**-53**	**51**
Run 333	**57**	**59**	**63**	**-61**	**-55**	**-49**
Run 334	**55**	**-53**	**-49**	**51**	**57**	**-63**
Run 335	**53**	**55**	**-51**	**-49**	**59**	**61**
Run 336	**51**	**-49**	**53**	**-55**	**61**	**-59**
Run 337	**49**	**51**	**55**	**53**	**63**	**57**
Run 338	**-47**	**45**	**41**	**43**	**33**	**-39**
Run 339	**-45**	**-47**	**43**	**-41**	**35**	**37**
Run 340	**-43**	**41**	**-45**	**-47**	**37**	**-35**
Run 341	**-41**	**-43**	**-47**	**45**	**39**	**33**
Run 342	**-39**	**37**	**33**	**-35**	**-41**	**47**
Run 343	**-37**	**-39**	**35**	**33**	**-43**	**-45**
Run 344	**-35**	**33**	**-37**	**39**	**-45**	**43**
Run 345	**-33**	**-35**	**-39**	**-37**	**-47**	**-41**
Run 346	**47**	**-45**	**-41**	**-43**	**-33**	**39**
Run 347	**45**	**47**	**-43**	**41**	**-35**	**-37**
Run 348	**43**	**-41**	**45**	**47**	**-37**	**35**
Run 349	**41**	**43**	**47**	**-45**	**-39**	**-33**
Run 350	**39**	**-37**	**-33**	**35**	**41**	**-47**
Run 351	**37**	**39**	**-35**	**-33**	**43**	**45**
Run 352	**35**	**-33**	**37**	**-39**	**45**	**-43**
Run 353	**33**	**35**	**39**	**37**	**47**	**41**
Run 354	**-31**	**29**	**25**	**27**	**17**	**-23**
Run 355	**-29**	**-31**	**27**	**-25**	**19**	**21**
Run 356	**-27**	**25**	**-29**	**-31**	**21**	**-19**
Run 357	**-25**	**-27**	**-31**	**29**	**23**	**17**
Run 358	**-23**	**21**	**17**	**-19**	**-25**	**31**
Run 359	**-21**	**-23**	**19**	**17**	**-27**	**-29**
Run 360	**-19**	**17**	**-21**	**23**	**-29**	**27**
Run 361	**-17**	**-19**	**-23**	**-21**	**-31**	**-25**
Run 362	**31**	**-29**	**-25**	**-27**	**-17**	**23**
Run 363	**29**	**31**	**-27**	**25**	**-19**	**-21**
Run 364	**27**	**-25**	**29**	**31**	**-21**	**19**
Run 365	**25**	**27**	**31**	**-29**	**-23**	**-17**
Run 366	**23**	**-21**	**-17**	**19**	**25**	**-31**
Run 367	**21**	**23**	**-19**	**-17**	**27**	**29**
Run 368	**19**	**-17**	**21**	**-23**	**29**	**-27**
Run 369	**17**	**19**	**23**	**21**	**31**	**25**

Run 370	**-15**	**13**	**9**	**11**	**1**
Run 371	**-13**	**-15**	**11**	**-9**	**3**
Run 372	**-11**	**9**	**-13**	**-15**	**5**
Run 373	**-9**	**-11**	**-15**	**13**	**7**
Run 374	**-7**	**5**	**1**	**-3**	**-9**
Run 375	**-5**	**-7**	**3**	**1**	**-11**
Run 376	**-3**	**1**	**-5**	**7**	**-13**
Run 377	**-1**	**-3**	**-7**	**-5**	**-15**
Run 378	**15**	**-13**	**-9**	**-11**	**-1**
Run 379	**13**	**15**	**-11**	**9**	**-3**
Run 380	**11**	**-9**	**13**	**15**	**-5**
Run 381	**9**	**11**	**15**	**-13**	**-7**
Run 382	**7**	**-5**	**-1**	**3**	**9**
Run 383	**5**	**7**	**-3**	**-1**	**11**
Run 384	**3**	**-1**	**5**	**-7**	**13**
Run 385	**1**	**3**	**7**	**5**	**15**

Layer1- Run 1 to Run 385
Layer2- Run 1 to Run 193
Layer3- Run 1 to Run 97
Layer4- Run 1 to Run 49
Layer5- Run 1 to Run 25
Layer6- Run 1 to Run 13

Run 370	**-15**	**13**	**9**	**11**	**1**	**-7**
Run 371	**-13**	**-15**	**11**	**-9**	**3**	**5**
Run 372	**-11**	**9**	**-13**	**-15**	**5**	**-3**
Run 373	**-9**	**-11**	**-15**	**13**	**7**	**1**
Run 374	**-7**	**5**	**1**	**-3**	**-9**	**15**
Run 375	**-5**	**-7**	**3**	**1**	**-11**	**-13**
Run 376	**-3**	**1**	**-5**	**7**	**-13**	**11**
Run 377	**-1**	**-3**	**-7**	**-5**	**-15**	**-9**
Run 378	**15**	**-13**	**-9**	**-11**	**-1**	**7**
Run 379	**13**	**15**	**-11**	**9**	**-3**	**-5**
Run 380	**11**	**-9**	**13**	**15**	**-5**	**3**
Run 381	**9**	**11**	**15**	**-13**	**-7**	**-1**
Run 382	**7**	**-5**	**-1**	**3**	**9**	**-15**
Run 383	**5**	**7**	**-3**	**-1**	**11**	**13**
Run 384	**3**	**-1**	**5**	**-7**	**13**	**-11**
Run 385	**1**	**3**	**7**	**5**	**15**	**9**

Layer1- Run 1 to Run 385
Layer2- Run 1 to Run 193
Layer3- Run 1 to Run 97
Layer4- Run 1 to Run 49
Layer5- Run 1 to Run 25
Layer6- Run 1 to Run 13

REFERENCES

Bingham, D., Sitter, R.R., Tang, B., (2009). Orthogonal and nearly orthogonal designs for computer experiments. *Biometrika*, **96 (1),** 51–65.

Cioppa, T. M. and Lucas, T. W. (2007). Efficient nearly orthogonal and spacefilling latin hypercubes, *Technometrics* **49(1),** 45–55.

Dam, E. R., Husslage, B., den Hertog, D. and Melissen, H. (2007). Maximin Latin hypercube designs in two dimensions. *Operations Research,* **55**, 158–169.

Dey, A. and Sarkar, D. (2014). A note on the construction of orthogonal latin hypercube designs, *Journal of Combinatorial Designs* 1–7. URL: http://dx.doi.org/10.1002/jcd.21408

Evangelaras, H. and Koutras, M. V. (2017). On second order orthogonal latin hypercube designs. *Journal of Complexity*, **39**:111–121

Georgiou, S.D., (2009). Orthogonal Latin hypercube designs from generalized orthogonal designs. *J. Statist. Plann. Inference,* **139 (4),** 1530–1540.

Georgiou, S.D., Efthimiou, I., (2014). Some classes of orthogonal latin hypercube designs. *Statist. Sinica*, **24**, 101–120.

Hickernell, F. J. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, **67**, 299–322.

Iman, R. L. and Conover, W. J. (1982). Distribution-free approach to inducing rank correlation among input variables. *Communications in Statistics:- Simulation and Computation*, **11**, 311–334.

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

Joseph, V. R. and Hung, Y. (2008). Orthogonal-maximin Latin hypercube designs. *Statistica Sinica,* **18**, 171–186.

Jin, R., Chen, W. and Sudjianto, A. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, **134**, 268–287.

Koehler, J.R., Owen, A.B. 1996. Computer experiments. In: Ghosh, S., Rao, C.R. (Eds.), *Handbook of Statistics*. Elsevier Science, New York, 261 –308.

Li, J. and Qian, P. Z. G. (2013). Construction of nested (nearly) orthogonal designs for computer experiments. *Statist. Sinica*, **23**, 451-466.

Liefvendahl, M. and Stocki, R. (2005). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference,* **136,** 3231 – 3247.

Lin, C. D., Bingham, D., Sitter, R. R. and Tang, B. (2010). A new and flexible method for constructing designs for computer experiments, *The Annals of Statistics,* 1460–1477.

Lin, C. D., Mukerjee, R. and Tang, B. (2009). Construction of orthogonal and nearly orthogonal latin hypercubes, *Biometrika,* **96(1),** 243–247.

Mandal, B. N., Dash, S., Parui, S., and Parsad, R. (2016). Orthogonal latin hypercube designs with special reference to four factors. *Statistics & Probability Letters*, **119**:181–185.

McKay, M. D., Beckman, R. J. and Conover, W. J. (1979). Comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics,* **21(2),** 239–245.

Morris, M. D. and Mitchell, T. J. (1995). Exploratory designs for computer experiments. *Journal of Statistical Planning and Inference,* **43**, 381–402.

Nguyen, N.-K. (2008). A new class of orthogonal latin hypercubes', *Statistics and Applications* **6**, 119–123.

Owen, A. B. (1992). A central limit theorem for Latin hypercube sampling. *Journal of the Royal Statistical Society, Series B (Methodological),* **54**, 541–551.

Pan, G., Ye, P. and Wang, P. (2014) A Novel Latin Hypercube Algorithm via Translational Propagation, *Scientific World Journal,* 1-15.

Pang, F., Liu, M.-Q. and Lin, D. K. (2009). A construction method for orthogonal latin hypercube designs with prime power levels, *Statistica Sinica*, **19(4),** 1721–1728.

Parui, S. (2015). Construction of Latin Hypercube Designs, ICAR-IASRI, Ph.D. Thesis Unpublished.

Parui, S., Mandal, B.N., Parsad, R., Dash, S. (2016). Orthogonal Latin hypercube designs for three columns. *Util. Math*. in press.

Qian, P. Z. (2009). Nested latin hypercube designs. *Biometrika*, **96(4),** 957–970.

Qian, P. Z. G., Ai, M. Y. and Wu, C. F. J. (2009). Construction of nested space-filling designs. *Ann. Statist.* **37**, 3616-3643.

Qian, P. Z. G., Tang, B. and Wu, C. F. J. (2009). Nested space-filling designs for computer experiments with two levels of accuracy. *Statist. Sinica*, **19**, 287-300.

QIAN, P. Z. G. and WU, C. F. J. (2008). Bayesian hierarchical modeling for integrating lowaccuracy and high-accuracy experiments. *Technometrics,* **50,** 192–204.

Qian, Z., Seepersad, C. C., Joseph, V. R., Allen, J. K., and Wu, C. J. (2006). Building surrogate models based on detailed and approximate simulations. J. Mech. Design, 128(4), 668–677.

Shannon, C. E. (1948). A mathematical theory of communication. The Bell System Technical Journal, 27, 379–423.

Steinberg, D. M. and Lin, D. K. (2006). A construction method for orthogonal latin hypercube designs, *Biometrika,* **93(2),** 279–288.

Sun, F., Liu, M.-Q. and Lin, D. K. (2009). Construction of orthogonal latin hypercube designs, *Biometrika*, **96**, 971–974.

Sun, F., Liu, M.-Q. and Lin, D. K. (2010). Construction of orthogonal latin hypercube designs with flexible run sizes, *Journal of Statistical Planning and Inference*, **140(11),** 3236–3242.

Sun, F. and Tang, B. (2017). A general rotation method for orthogonal latin hypercubes. *Biometrika*, **104(2)**, 465–472.

Sun, F. S., Yin, Y. H., and Liu, M. Q. (2013). Construction of nested space-filling designs using difference matrices. *J. Statist. Plann. Inference*, **143**, 160-166.

Tang, B. (1993). Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association,* **88**, 1392–1397.

Tang, B. (1994). A theorem for selecting OA-based Latin hypercubes using a distance criterion. *Communications in Statistics: Theory and Methods*, **23**, 2047–2058.

Viana, F. A. C., Venter, G. and Balabanov, V. (2010). An algorithm for fast optimal Latin hypercube design of experiments. *International Journal for Numerical Methods in Engineering,* **82**, 135–156.

Yang, J. and Liu, M.-Q. (2012). Construction of orthogonal and nearly orthogonal latin hypercube designs from orthogonal designs, *Statist. Sinica* **22**, 433–442.

Ye, K. Q. (1998). Orthogonal column latin hypercubes and their application in computer experiments, *Journal of the American Statistical Association*, **93(444),** 1430–1439.

Ye, K. Q., Li, W. and Sudjianto, A. (2000). Algorithmic construction of optimal symmetric Latin hypercube designs. *Journal of Statistical Planning and Inference*, **90**, 145–159.

Yang, J. and Liu, M. Q. (2012). Construction of orthogonal and nearly orthogonal Latin hypercube designs from orthogonal designs. *Statistica Sinica*, **22,** 433-442.

Yang, J., Liu, M. Q. and Lin, D. K. J. (2014). Construction of Nested Orthogonal Latin Hypercube Designs, *Statistica Sinica,* **24,** 211-219

Zhu, H., Liu, L., Long, T. and Peng, L. (2011). A novel algorithm of maximin Latin hypercube design using successive local enumeration. *Engineering Optimization,* **1**, 1–14.