

N-ary Codes, Designs, and Enclosings

By

Alexander L. Strehl

Advised by

Dr. Dinesh G. Sarvate

Department of Mathematics
Department of Computer Science
College of Charleston, SC

This essay contains original research within discrete mathematics. Material from this essay has also been presented in a mathematics presentation, a poster session, and two research papers. One has been accepted for publication and the second has been submitted and is being considered for publication:

1) *Linear Codes Through Latin Squares Modulo n* by Dinesh G. Sarvate and Alexander L. Strehl has been accepted for publication in *Bulletin of the Institute for Combinatorics and Its Applications*

2) *A Few Results on Ternary Codes from Ternary Designs* by Dinesh G. Sarvate and Alexander L. Strehl has been submitted for publication.

Table of Contents

| | | |
|---|--|----|
| 1 | Chapter 1. Introduction | 3 |
| | 1.1 Discrete Mathematics | 3 |
| | 1.2 Coding Theory | 5 |
| | 1.3 Latin Squares | 8 |
| | 1.4 Enclosings | 10 |
| 2 | Chapter 2. Ternary Codes Through Ternary Designs | 14 |
| | 2.1 Balanced Incomplete Block Designs | 14 |
| | 2.2 Balanced Ternary Designs | 16 |
| | 2.3 Results on Ternary Codes Through Ternary Designs | 18 |
| | 2.4 Examples | 24 |
| 3 | Chapter 3. Linear Codes Through Latin Squares ... | 27 |
| | 3.1 Introduction and Basic Definitions | 27 |
| | 3.2 Results | 29 |
| | 3.3 Examples | 35 |
| | References | 37 |
| | Appendix A (LISP Program) | 38 |

Chapter 1. Introduction

1.1. Discrete Mathematics

The aim of this essay is to study the interconnection between certain topics within discrete mathematics. Specifically, this paper deals with n -ary codes, designs, and enclosings. It is not the goal of this paper to survey or discuss every aspect of n -ary codes, designs, and enclosings. Instead we will focus on some specific areas within this topic in an effort to solve interesting problems and research new areas.

There are many different viewpoints concerning the meaning of the term discrete mathematics. One simple definition is that discrete mathematics is the study of the properties of different arrangements of finitely many objects. This definition would imply that discrete mathematics deals with permutations and combinations of finite sets with specific properties. These areas are then very closely related to the binomial theorem and the many-fold identities that come from it. Thus, it would follow that the binomial identities are part of discrete mathematics as well. However, this definition also encompasses the theory of finite graphs, which is really just the study of the different ways in which a finite number of objects can be related or connected. These are all mathematical areas that most would accept as discrete. Yet, it is certainly clear that many more mathematical concepts can be justified as discrete under this definition. For example, the study of partially ordered sets of finite cardinality satisfying certain other properties (lattices for example), or for that matter the study of finite groups, finite fields, and more generally, finite sets. Furthermore, recurrence relations often show up during the study of these structures, thus they are also usually classified under discrete mathematics. So far everything that we have classified as discrete deals with finite quantities. However, in many cases the study of certain finite collections of objects is very much related to the study of their infinite analogues. Thus, for example, it seems perfectly valid to classify the study of all Boolean algebras as discrete rather than just the Boolean algebras of finite cardinality. Yet, all of the areas within mathematics that deal with infinite quantities of objects can be simplified to finite cases, so what stops us from classifying all of mathematics as discrete?

Another way to see things is to view discrete mathematics as opposed to analysis, which deals with continuous (rather than discrete) quantities. Between any two real numbers there are infinitely many real numbers. This is one of the properties of real numbers that justify the name continuous. Due to this property of continuous number systems, one can find a sequence of numbers that keep getting closer and closer to another number (and never stop getting closer), but never reach that number. For example the sequence $\{1/1, 1/2, 1/3, 1/4, \dots\}$ approaches zero but never reaches zero. This is not always the case. For example, consider the set of ordinary integers. Between any two distinct integers there are only finitely many integers. Therefore, if a sequence of integers gets closer and closer to some number, it will eventually reach that number every time. This distinction between the real number system and the integers can be helpful to illustrate the main differences between analysis and discrete mathematics, however it does not tell the whole story. The main field of mathematics that studies and attempts to describe the integers is number theory. Discrete mathematics is a separate, although related field and studies

mathematical objects that are in some sense discrete rather than continuous.

Perhaps a better way to describe the subject of discrete mathematics, or combinatorics, is to describe how it is done rather than to directly define it. Two important things that are almost always encountered when doing discrete mathematics are the problem of existence and the problem of how many. When a new combinatorial structure is being studied, the first question to ask is when does an instance of this structure (often called an object) exist. If existence of the structure under question is the norm, we tend to ask under what conditions is it the case that no instances of some discrete structure exist. For example, when latin squares of order six are considered, we cannot find two that are mutually orthogonal, but it is interesting that this situation never occurs for any orders larger than 6. Once it is determined when some discrete structure exists, the next question is how many such distinct non-isomorphic objects can we find. To solve both of these problems the discrete mathematician often utilizes methods of construction and techniques of counting. It is arguably true that the most effective foolproof way of insuring the existence of some structure is to construct an instance of that structure (and this area of discrete mathematics ties nicely into the design, development, and analysis of algorithms). Also, it is often the case that when constructing an instance of a discrete structure, one must try many different combinations and arrangements of some finite set of symbols or objects. This is often the justification for the previous definition of discrete mathematics as the study of the properties of different arrangements of finitely many objects. However, any adequate combinatorial toolbox will include counting principles right alongside construction methods. Many times it happens that one method of construction can produce a large number of non-isomorphic discrete objects, and it then becomes the job of the discrete mathematician to determine the exact number of these. Of course, there is much more to combinatorics than what has been mentioned; many other questions about these structures are asked. For example, one is often interested in the problem of embedding, enclosing, and generalizing, among others.

Although counting arguments appear in every branch of mathematics, they play a much more fundamental role in combinatorics. A simple combinatorics problem is to count the number of possible combinations of some number of objects. If there are five objects, there are ten combinations that contain three of the five objects. For example, if the five objects are symbolized as $\{a, b, c, d, e\}$, then the ten combinations that choose three objects could be described as

$$\{a, b, c\}, \{a, b, d\}, \{a, b, e\}, \{a, c, d\}, \{a, c, e\}, \{a, d, e\}, \{b, c, d\}, \{b, c, e\}, \{b, d, e\}, \{c, d, e\}.$$

In general, if we want to know the total number of combinations of k objects chosen from a set of n objects, we can use the formula $(n!)/((n - k)! k!)$. This is a typical example of the end result of a counting argument. Counting arguments proliferate in discrete mathematics but are also used in every branch of mathematics. For example, a famous theorem of group theory, called Lagrange's theorem, basically counts the elements of the cosets of a group to prove that the order of any subgroup divides the order of the group it is contained within. Groups are the main structures of algebra, an enormously exciting branch of mathematics. Basically a group is any set of objects (for example the real numbers) under some binary operation (for example addition) that satisfy some basic

properties (for example the associative law). By studying the laws inherent in this concept of an abstract group, one can prove results simultaneously for all such systems that satisfy the group axioms. This is a tremendously powerful form of abstraction and is used to prove results for discrete as well as continuous mathematical systems.

Besides being interesting in its own right, discrete mathematics is also known to be have especially fruitful applications to many fields of study. The results and techniques of discrete mathematics have been used in geometry, algebra, physics, the theory of computation, artificial intelligence, and the design of circuits, among others. It has been most useful, perhaps, in the study of computer science, being one of the most important tools a computer scientist possesses.

Although there are many distinctions between the different types of quantities and techniques used in discrete mathematics and those of other branches of mathematics, these distinctions are in many ways artificial. All of mathematics is related by several things, and all of mathematics is governed by logic and reason. There are no strict borders from one mathematical field to another and there are many areas where substantial overlap occurs. The results from one branch of mathematics are often fundamentally linked with those of another. In some instances, it is really a case of seeing two sides of the same coin. Furthermore, each branch of mathematics is often further divided into sub-branches or categories. Combinatorics is not exempt from this occurrence and itself has many huge sub-fields of study. In this essay we will look at topics from some of these sub-fields and discover how it is often the case that these sub-fields are interrelated.

1.2. Coding Theory

Coding theory is a branch of mathematics that is also important to computer science. It deals with the transmission of information across some line of communication. The goal is to encode the information in an intelligent manner so that if errors occur during the transmission stage, these errors can be detected and/or corrected by the receiver. Because of its application to computer science most work in this field has been done under the assumption that the codes will always be made of zeros and ones (called binary codes). Less work has been done with n -ary codes, the generalization of binary codes. The following definitions are some of the most basic to all of coding theory.

Definition 1.1. A **codeword** is some string made up of elements from some given finite set A (called the **alphabet**). A **code** is a set of codewords of length n .

Example 1.2. If $A = \{a, b, c\}$, then one possible code C is $C = \{aab, cab, aba, cba\}$.

In Example 1.2, we can see that C is a code made up of four codewords. We can also see that within a codeword only symbols from the set A appear. It is also clear that the symbols within a codeword can be repeated and that the order of the symbols

has importance. This is because a **string** is defined as an ordered list of symbols. This concept can be opposed to that of a set.

When it is said that the objects included in a set are distinct, it is meant that a set contains only one copy of each element, and duplicates are not counted. That is one difference between a set and a string. Another is that the objects, called elements, of a set need not have any order, whereas the symbols of a string are necessarily ordered. In Definition 1.1, A is a set and the curly braces indicated the start and close of the listing of elements within the set A . It can be seen that the elements of A are listed in a specific order, namely a, b, c . However, this ordering is arbitrary and only necessary for the purpose of communication of the elements of A in this essay. In many instances, a set can be defined by a rule, such as $S = \{x \mid x \text{ is an english word}\}$. In this case, S is the set of all elements x , where x is an english word.

Notice that we could have used the set $B = \{0, 1, 2\}$ in place of the set A in Example 1.2. Then let's define $D = \{001, 201, 010, 210\}$. We can see that this new D is in some sense "the same" as the old C where a is renamed 0, b is renamed 1, and c is renamed 2. To indicate this in mathematical terminology we say that there exists a **bijection** between the sets C and D . It is clear that for any finite set S there is a bijection (one-to-one correspondence) between S and the set Z_n where n is the order of the set S and $Z_n = \{0, 1, 2, \dots, n - 1\}$. Therefore, from now on we will only consider codes made up of elements from Z_n .

Since we are interested in transmitting and receiving codewords we need to first determine a scheme that allows the detection of errors introduced during the transmission of the codewords. One fundamental assumption of most of coding theory, is that errors don't occur very often. That is, we assume that the message which is received is usually the same as the message that was sent. This assumption is a good one, because a certain line of communication is usually used only if there is a great chance of receiving the message that was intended to be sent. Therefore, when a string is received (which we will from now on call the received vector) that is not part of the code, so that we know an error has occurred during the transmission, we often correct the received string to the codeword that most resembles the received string. This error-correction policy is called the "nearest neighbor" method. With this in mind, we define distance so that we can correct received messages to code words in such a way as to minimize their distance.

Definition 1.3. *The **distance** between two codewords x and y of length n is the number of positions in which x and y differ, and is denoted $d(x, y)$.*

Example 1.4. *Suppose that $x = (0, 1, 3, 2, 5, 1, 0, 2, 2)$ and $y = (4, 4, 3, 1, 5, 1, 2, 2, 1)$ are two n -ary (for $n \geq 6$) codewords of length 9. Then, the distance between x and y is 5.*

A code is just a set of codewords of length n . It is therefore possible to have a binary operation on this code. If the code is closed under this binary operation, we call it a linear

code.

Definition 1.5. Suppose C is a code with a specified binary operation $*$. Then, C is **linear** if for any elements $a, b \in C$ it is the case that $a * b \in C$.

Another way to say this is that a code C is linear iff C is a groupoid with respect to the given binary operation $*$. This property is often useful when one wishes to correct a received string. A nice way to specify the operation for a code is to do so indirectly. First, a binary operation is defined for the alphabet over which the code is defined. Then a code of length n can be viewed as a subset of the cross product of the alphabet with itself n times. This is useful because any code defined over this alphabet now has a natural binary operation. We are also interested in looking at all the distances between all the pairs of codewords in a given code. This leads us to the following definition

Definition 1.6. The **minimum distance** or **distance** of a code C is the minimum of the set $\{d(x, y) \mid x, y \in C\}$

1.3. Latin Squares

One combinatorial structure that has been studied for a long time is the latin square. Latin squares are very interesting from a combinatorial viewpoint and they have many applications. This is perhaps what makes them so popular.

Definition 1.7. A **latin square** is an $n \times n$ matrix (or array) of n symbols such that each row and column of the matrix is a permutation of these n symbols

One important attribute of a latin square is that each element appears once and only once in each row or column. Algebraists will note that the multiplication table of a group exhibits this property and is therefore a latin square if you remove the bordering. However, the converse is not true. There exist latin squares that cannot represent the unbordered multiplication table of a group. We will see, however, that correct algebraic structure to associate with a latin square is the quasigroup.

Definition 1.8. If S is a set with a binary operation $*$, then S is a **quasigroup** if for every pair of elements $y, z \in S$ there exists unique (but perhaps equal) elements $x_1, x_2 \in S$ such that $x_1 y = z$ and $y x_2 = z$.

A group is just an associative quasigroup with identity. Therefore, every group is a

quasigroup but not every quasigroup is a group. It is easy to see that the two concepts latin square and (finite) quasigroup are really one in the same. This is a very deep connection between algebra and combinatorics (although some feel that algebra should be classified as combinatorics this is not always the case). If you border a latin square so that row i and column j correspond with element y then for any fixed element z appearing in the latin square the equations $xy = z$ and $yx = z$ each have one and only one solution because z appears exactly once in the i th row and j th column. Similarly if S is any finite quasigroup, and y and z are fixed elements of S , we can conclude from the fact that $xy = z$ and $yx = z$ each have one and only one solution, that the element z appears once and only once in the column and row corresponding to y in the multiplication table of S . From the fact that z was chosen arbitrarily, we can conclude that the column and row corresponding to y are permutations of the n elements of S . However, y was also chosen arbitrarily, so we conclude that every row and column of the multiplication table of S is a permutation of the n elements of S and therefore a latin square.

Example 1.9. *The following are examples of latin squares*

$$\begin{bmatrix} 5 & 3 & 1 & 2 & 4 \\ 2 & 4 & 3 & 1 & 5 \\ 3 & 1 & 4 & 5 & 2 \\ 1 & 2 & 5 & 4 & 3 \\ 4 & 5 & 2 & 3 & 1 \end{bmatrix}, \begin{bmatrix} a & b & c & f & e & d & g \\ g & d & e & c & f & b & a \\ c & f & d & a & g & e & b \\ f & e & a & d & b & g & c \\ d & g & f & b & c & a & e \\ b & c & g & e & a & f & d \\ e & a & b & g & d & c & f \end{bmatrix}$$

Now let's consider these two small 3×3 latin squares made up of the same symbols:

$$L_1 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}, L_2 = \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

Notice that we can get pairs of numbers from these two latin squares as follows: For each position i, j where $i = 0, 1, 2$ and $j = 0, 1, 2$ we can form the pair (x_{ij}, y_{ij}) where x_{ij} is the (ij) th position of the first latin square and y_{ij} is the (ij) th position of the second latin square. If we do this we get the set

$$\{(0, 0), (1, 1), (2, 2), (1, 2), (2, 0), (0, 1), (2, 1), (0, 2), (1, 0)\}$$

It is clear that we can perform this operation on any two latin squares of the same size and it is also clear that if the latin squares are of order n , then we get at most n^2 distinct pairs. If we get n^2 distinct pairs then we say that these two latin squares are **mutually orthogonal**. The next question is for any given n , what is the size of a maximal set of latin squares such that every pair of latin squares in the set are mutually orthogonal. This

question has brought about a great deal of research. What is important in our study is the connection between the theory of mutually orthogonal latin squares and that of coding theory. There is a very strong connection, because Denes and Keedwell [1] have shown a method of constructing an error-correcting code of distance $t + 1$ with n^2 codewords of length $t + 2$ when given t mutually orthogonal latin squares. Their method is as follows:

Given t mutually orthogonal latin squares L_1, L_2, \dots, L_t , the code is the set of all codewords of the form $(i, j, l_1, l_2, \dots, l_t)$ where l_1 is the i, j -th entry of L_1 , l_2 is the i, j -th entry of L_2 , and l_k is the i, j -th entry of L_k where $1 \leq k \leq t$.

This method of construction was also rediscovered by Kadowaki, Kageyama, Kimura, and Yanagida [2]. In particular they have shown that the code resulting from the latin squares L_1 and L_2 is perfect (every vector is correctable). Let's examine the code generated by this method when we take L_1 and L_2 as our set of mutually orthogonal latin squares:

$$\{(0, 0, 0, 0), (0, 1, 1, 1), (0, 2, 2, 2), (1, 0, 1, 2), (1, 1, 2, 0), \\ (1, 2, 0, 1), (2, 0, 2, 1), (2, 1, 0, 2), (2, 2, 1, 0)\}$$

A noteworthy feature of this code is that it is also a linear code when addition and multiplication are defined modulo n . The question that arises naturally is when, if ever, do other linear codes result from this construction? When a code C results from applying the above method to some set of mutually orthogonal latin squares we say that these latin squares **generate** C . If C is a linear code we say that these latin squares **generate a linear code modulo n** , where n is the order of the latin squares. In this essay we explore this connection between latin squares and coding theory and determine exactly when a linear code results from this method of construction.

1.4. Enclosings

When one encounters a new (or old) combinatorial structure there are generally several questions to ask. Firstly, there is the problem of existence. Under which conditions does an instance of this structure exist? The problem of the existence of mutually orthogonally latin squares is an example of this. The next question to ask is how many of these objects exist. This question for latin squares has been studied, however once the latin square becomes big the total number of latin squares becomes so large it is hard to even estimate the exact number. After these main questions have been studied ones turns to the problem of enclosings: Under which conditions can one instance of the structure be enclosed within another larger instance of the structure. In reference to latin squares, this is the question of when is it possible to take a given $n \times n$ latin square and make it a part of a larger latin square. More formally, an enclosing of a latin square is an $n \times n$ latin square, such that the removal of $r > 1$ rows, and corresponding columns, results in a latin square.

Theorem 1.10. *If L is an $n \times n$ latin square enclosed within another latin square M , then M has order greater than or equal to $2n$.*

Proof. Let m be the order of the latin square M and n be the order of the latin square L . Now, since M encloses L , there is a column of M that can be removed, along with some other columns and rows, so that we are left with L . This column has m elements. There are n rows of M that include the rows of L . So m has n positions, all of which are in the rows that include L . Hence, these m positions cannot be filled with elements from L , since they are already used in the rows of these positions. Thus, these n positions must be filled with distinct elements not from L . Therefore, we see that M must have at least $2n$ elements. \square

Theorem 1.11. *If L is an $n \times n$ latin square, there exists a $2n \times 2n$ latin square M that is an enclosing of L .*

Proof. Let L' be latin square of order n obtained directly from L by replacing each of the symbols of L by distinct symbols not found in L . Then form the block array:

$$\begin{bmatrix} L & L' \\ L' & L \end{bmatrix}$$

Each row(column) of this new array consists of one row(column) of L and one row(column) of L' . Since L and L' are both latin squares and have no common symbols among them, we have that this new array is also a latin square. Clearly, from it's structure, we can see that it is an enclosing of L and L' . \square

We know that an enclosing of a latin square must at least twice the size of the original latin square. The next question would be whether there exists enclosings of more than

twice the size of the original latin square. Some examples will indicate that this is the case.

Examples. Here are examples of two enclosings of a 3×3 latin square made up of the elements $\{a, b, c\}$. The first is an enclosing within a latin square of order 7 and the second is an enclosing within a latin square of order 8. The original latin square being enclosed, always appears as the intersection of the first three rows and columns of these examples. For the sake of the examples, a specific latin square of order 3 has been chosen, although this could be replaced with any latin square of order 3 without upsetting the larger latin square.

$$\begin{bmatrix} a & b & c & 1 & 2 & 3 & 4 \\ b & c & a & 2 & 3 & 4 & 1 \\ c & a & b & 3 & 4 & 1 & 2 \\ 1 & 2 & 3 & 4 & a & b & c \\ 2 & 3 & 4 & a & 1 & c & b \\ 3 & 4 & 1 & b & c & 2 & a \\ 4 & 1 & 2 & c & b & a & 3 \end{bmatrix}, \begin{bmatrix} a & b & c & 1 & 2 & 3 & 4 & 5 \\ b & c & a & 2 & 3 & 4 & 5 & 1 \\ c & a & b & 3 & 4 & 5 & 1 & 2 \\ 1 & 2 & 3 & 5 & a & b & c & 4 \\ 2 & 3 & 4 & a & 5 & 1 & b & c \\ 3 & 4 & 5 & b & 1 & c & 2 & a \\ 4 & 5 & 1 & c & b & 2 & a & 3 \\ 5 & 1 & 2 & 4 & c & a & 3 & b \end{bmatrix}$$

Theorem 1.12. If L is an $n \times n$ latin square, there exists a $(2n + 1) \times (2n + 1)$ latin square M that is an enclosing of L .

Proof. It is well known that there always exists a latin square with a traversal for any given n . Therefore, we can find a latin square $K = \| k_{ij} \|$ of order n , that has distinct values on the lower diagonal. We also insist that the elements of K be the same as the elements of L . Now we will construct a latin square M of order $(n + 1) \times (n + 1)$ based upon K . Firstly, set the diagonal elements of M to $m_{ii} = i$. Now if $i > 0$, $j < n$, and $i \neq j$, then let $m_{ij} = k_{i(j-1)}$. Thus we have defined all of M except the non-diagonal elements of the first column and last row. We note that the i th row of M contains all of the elements of the i th row of K except for the that element of K which is part of the lower diagonal (if such an element exists). Hence in all the remaining positions of the first column of M (except for the last position) we will use this element from the lower diagonal of K . That is for m_{i0} with $1 \leq i \leq (n - 1)$ let $m_{i0} = k_{i(i-1)}$. Similarly for the last row, let $m_{nj} = k_{j(j-1)}$ when $1 \leq j \leq (n - 1)$. Now we have defined every position except for m_{n0} . However, we note that the $n - 1$ elements from K of the first column of M are just those elements on the lower diagonal of K , which are given to be distinct, hence there is one remaining element from K , say x . Thus, we let $m_{n0} = x$. Now consider the $n \times (n + 1)$ array $N = \| n_{ij} \|$, defined by $n_{ij} = (i + j + 1) \text{ mod } (n + 1)$. Let the embedding of L be of the form:

$$E = \begin{bmatrix} L & N \\ N^T & M \end{bmatrix}$$

We claim that this is a latin square. We have already shown that every row and column of E contains the n elements of L . Furthermore it is clear that the i th column of N (same as the i row of N^T) has the all the integers between 0 and n except for i . However in the $(n + i)$ th column of E (which contains the i th column of N) the element i occurs in the $(n + i)$ th row (corresponding to m_{ii}). Thus, E is a latin square containing L , and we are done. \square

The proof of Theorem 1.12 involves a construction. This is often the case in combinatorics when one wants to prove that a certain structure, such as an enclosing, exists. It is sometimes hard, if not impossible, to really understand and comprehend a constructional proof without an example. Therefore, let us consider the problem of enclosing the following latin square of order 8 in a latin square of order 17:

$$L = \begin{bmatrix} A & B & C & D & E & F & G & H \\ B & C & D & E & F & G & H & A \\ C & D & E & F & G & H & A & B \\ D & E & F & G & H & A & B & C \\ E & F & G & H & A & B & C & D \\ F & G & H & A & B & C & D & E \\ G & H & A & B & C & D & E & F \\ H & A & B & C & D & E & F & G \end{bmatrix}$$

Our first step to find another latin square of order 8 that has no repeated elements on its lower diagonal. If L satisfied this property we could let $K = L$. However the lower diagonal of L is $BDFHBDF$, which clearly contains repeated elements. So we will use the following latin square K that has no repeat elements on its lower diagonal. Such latin squares can be obtained easily from any latin square (of the correct size) with a traversal by changing the order of the columns so that the elements of the traversal are moved to the lower diagonal:

$$K = \begin{bmatrix} C & B & D & G & A & E & F & H \\ A & H & B & E & G & C & D & F \\ H & G & A & D & F & B & C & E \\ G & F & H & C & E & A & B & D \\ F & E & G & B & D & H & A & C \\ E & D & F & A & C & G & H & B \\ D & C & E & H & B & F & G & A \\ B & A & C & F & H & D & E & G \end{bmatrix}$$

Now the remaining steps of the construction process yield

$$E = \begin{bmatrix} A & B & C & D & E & F & G & H & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 0 \\ B & C & D & E & F & G & H & A & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 0 & 1 \\ C & D & E & F & G & H & A & B & 3 & 4 & 5 & 6 & 7 & 8 & 0 & 1 & 2 \\ D & E & F & G & H & A & B & C & 4 & 5 & 6 & 7 & 8 & 0 & 1 & 2 & 3 \\ E & F & G & H & A & B & C & D & 5 & 6 & 7 & 8 & 0 & 1 & 2 & 3 & 4 \\ F & G & H & A & B & C & D & E & 6 & 7 & 8 & 0 & 1 & 2 & 3 & 4 & 5 \\ G & H & A & B & C & D & E & F & 7 & 8 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ H & A & B & C & D & E & F & G & 8 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 0 & C & B & D & G & A & E & F & H \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 & 0 & A & 1 & H & B & E & G & C & D & F \\ 3 & 4 & 5 & 6 & 7 & 8 & 0 & 1 & G & H & 2 & A & D & F & B & C & E \\ 4 & 5 & 6 & 7 & 8 & 0 & 1 & 2 & H & G & F & 3 & C & E & A & B & D \\ 5 & 6 & 7 & 8 & 0 & 1 & 2 & 3 & B & F & E & G & 4 & D & H & A & C \\ 6 & 7 & 8 & 0 & 1 & 2 & 3 & 4 & C & E & D & F & A & 5 & G & H & B \\ 7 & 8 & 0 & 1 & 2 & 3 & 4 & 5 & F & D & C & E & H & B & 6 & G & A \\ 8 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & E & B & A & C & F & H & D & 7 & G \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & D & A & G & H & B & C & F & E & 8 \end{bmatrix}$$

Although the construction of the proof of Theorem 1.12. is new, there is a much stronger result known. The following Theorem has been proved by T. Evans [3]:

Theorem 1.13. *For given integers n and k , n arbitrary and $k \leq n/2$ there exists a quasigroup of order n which contains at least one subquasigroup of order k .*

Of course we have seen that once one latin square of order k is embedded in another latin square of order n , we can then embed any latin square of order k in some latin square of order n .

Chapter 2. Ternary Codes Through Ternary Designs

2.1. Balanced Incomplete Block Designs

The type of designs considered in this chapter can all be described by special matrices called incidence matrices. The incidence matrices of ternary designs are made up of three symbols. Ternary codes are also made up of three symbols. In this essay we always take these three symbols to be $(0, 1, 2)$. Binary codes, on the other hand, are codes over an alphabet of only two symbols $(0, 1)$. The term used for a certain type of design that can be described by an incidence matrix using only two symbols $(0, 1)$ is balanced incomplete block design. Before attempting to develop a theory for ternary designs and some relationships to ternary codes it is best to understand some elementary facts about balanced incomplete block designs and how they might be used to develop some binary codes. After these are understood, we will focus on generalizing the results to work with ternary codes and ternary designs. It is important to mention that ternary designs generalize balanced incomplete block designs.

Definition 2.1. A balanced incomplete block design, denoted by (v, b, r, k, λ) is a finite collection B of b subsets (called blocks) of some finite set S with v elements, such that each block has k elements, each element of S appears in r blocks, and each pair of distinct elements from S appears in λ blocks.

Example 2.2. Let $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$. Now let the following be blocks: $B_1 = \{s_2, s_4, s_5\}$, $B_2 = \{s_1, s_4, s_7\}$, $B_3 = \{s_2, s_6, s_7\}$, $B_4 = \{s_1, s_2, s_3\}$, $B_5 = \{s_3, s_5, s_7\}$, $B_6 = \{s_3, s_4, s_6\}$, $B_7 = \{s_1, s_5, s_6\}$. Then we have a balanced incomplete block design (BIBD) with parameters $(7, 7, 3, 3, 1)$. This design can be created using Hadamard matrices (for example, see [7]).

It is useful to illustrate a BIBD with a matrix where the rows are labeled with elements from S and the columns are labeled with elements from B . Then, a given entry (i, j) of the matrix has a 1 if the element corresponding to the i th row is a member of the block corresponding to the j th column. If the entry does not get a 1, then it gets a 0. We say that this matrix is the **incidence matrix** of the given BIBD. The matrix will have v rows and b columns. The incidence matrix for the design given in Example 2.2 is

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

We know from the definition of a balanced incomplete block design that in any incidence matrix each row will contain r ones, each column will contain k ones, and there will be λ columns that contain a one in any two fixed rows. Now, we will obtain one of the most simple (and fundamental) theorems of balanced incomplete block designs by using a counting argument. Let's count the total number of ones in the incidence matrix in two ways, first from row to row, and then from column to column. We know there are r ones in each row and that there are v rows, hence there is a total of vr ones in the incidence matrix. On the other hand, we know there are k ones in each column and that there are b total columns. Thus, there are bk ones in the incidence matrix. We have shown the truth of part of the following theorem.

Proposition 2.3. *If (v, b, r, k, λ) are the parameters for some balanced incomplete block design, then $vr = bk$ and $\lambda(v - 1) = r(k - 1)$.*

In Fujitake, Kageyama, and Shimata [5], it was shown under which conditions a binary error-correcting code results from the following structure:

$$\begin{bmatrix} \vec{0}_b \\ N \\ J_{vxb} - N \\ \vec{1}_b \end{bmatrix}$$

where N is the incidence matrix of a BIBD, and $J_{vxb} = (\vec{1}_v)^T \vec{1}_b$. In this essay we discuss some necessary conditions on ternary designs such that the incidence matrix of a ternary design gives a ternary error-correcting code under the same structure. Adopting the notation from Fujitake, Kegeyama, and Shimata [5], we call the above structure $*$. Basically the result for BIBDs is that the structure $*$ produces at least a one error-correcting code except in some trivial cases. For example, lets consider the structure $*$ using Example 2.2:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Since the minimum distance of this code is greater than or equal to 3, we have that it is at least a one error-correcting code. Many methods of constructing BIBDs have been discovered, and this result tells us that almost all of these BIBDs can also be used to form nontrivial binary error-correcting codes. Our next goal will be to generalize this result to ternary codes. At first this seems like a straightforward concept but we will quickly find that it becomes more difficult when dealing with three underlying symbols rather than two (which suggests that an even more general result over n symbols is out of reach at this time).

2.2. Balanced Ternary Designs

A **Balanced Ternary Design** or BTDesign is defined the same way as a BIBD, however elements may appear twice within a given block. This means that the blocks are no longer sets but are now multisets, limited by the fact that no element may appear more than twice, hence the term "ternary". More formally, we say that a BTDesign has parameters $(V, B; \rho_1, \rho_2, R; K, \Lambda)$ if S has V elements such that the following properties hold: (1) each element appears only once in ρ_1 blocks and twice in exactly ρ_2 blocks; (2) There are B blocks containing K elements; (3) Each element appears R times altogether; (4) Each pair of elements appears Λ times. Now we will prove some well-known elementary properties of BTDesigns.

Proposition 2.4. *If a BTDesign has parameters $(V, B; \rho_1, \rho_2, R; K, \Lambda)$ then $VR = BK$.*

Proof. There are V elements and each element occurs R times altogether, therefore the order of the set $C = \{(v, b) \mid \text{element } v \text{ occurs in block } b\}$ is VR . On the other hand, there are B blocks, and each contains K total elements (where duplicates are counted), hence we have that the order of C is also BK . We have shown that $|C| = VR = BK$. \square

Proposition 2.5. *If a BTDesign has parameters $(V, B; \rho_1, \rho_2, R; K, \Lambda)$ then $R = \rho_1 + 2\rho_2$*

Proof. Each element of the BTDesign occurs R times altogether. However, we also know that each element occurs ρ_1 times singly and ρ_2 doubly. This means that for any fixed element x in the BTDesign occurs once in ρ_1 blocks and twice in ρ_2 blocks. Clearly we have that x occurs a total of $\rho_1 + 2\rho_2 = R$ times altogether. \square

Proposition 2.6. *If a BTDesign has parameters $(V, B; \rho_1, \rho_2, R; K, \Lambda)$ then $\Lambda(V - 1) = \rho_1(K - 1) + 2\rho_2(K - 2) = R(K - 1) - 2\rho_2$*

Proof. Fix an element x in the BTDesign. We want to count the number of pairs containing x that occur in any block; that is we want to count the elements of the set $C = \{(x, y) \mid x, y \in b, \text{ for some block } b\}$. Now from the definition of a Balanced Ternary Design we are given that each pair of elements occurs Λ times, and there are $V - 1$ elements, not counting x ,

so we have that $|C| = \Lambda(V - 1)$, since x appears in Λ pairs with each of the remaining $V - 1$ elements. Now, we also know that x occurs singly in ρ_1 blocks. Hence, in ρ_1 blocks x occurs with $k - 1$ elements (since each block has a total of K elements). Also, in ρ_2 blocks two copies of x appear with $K - 2$ other elements (they may not all be distinct). Therefore we have that $|C| = \rho_1(K - 1) + 2\rho_2(K - 2)$. Finally, using the fact that $R = \rho_1 + 2\rho_2$, we can see that

$$\begin{aligned} R(K - 1) - 2\rho_2 &= (\rho_1 + 2\rho_2)(K - 1) - 2\rho_2 = \rho_1(K - 1) + 2\rho_2(K - 1) - 2\rho_2 \\ &= \rho_1(K - 1) + 2\rho_2(K - 2) = \Lambda(V - 1). \end{aligned}$$

□

The first thing to notice is that any results which hold for Balanced Ternary Designs also hold for Balanced Incomplete Block Designs (when they make sense). This is because BTDs generalize BIBDs. So we can see that the truth of Proposition 2.4 ensures the truth of Proposition 2.3. Once we prove Proposition 2.4 it is redundant to prove Proposition 2.3. In BIBDs, elements are not allowed to appear more than once in any block, so any BIBD can be viewed as a BTM where $\rho_2 = 0$, and therefore, $R = \rho_1$. From Proposition 2.4, we conclude that the parameters (v, b, r, k, λ) , for any BIBD satisfy $\lambda(v - 1) = r(k - 1)$. This is much simpler than the statement of Proposition 2.6, because with BIBDs the terms with ρ_2 vanish. It is clear that even with these elementary results, the situation is much more complicated when BTDs rather than BIBDs are considered.

Many good examples of Balanced Ternary Designs can be found in a paper entitled "A List of Balanced Ternary Designs with $R \leq 15$, and Some Necessary Existence Conditions" by Elizabeth J. Billington and Peter J. Robinson [2]. Although constructing a particular BTM does not prove a general result (it does, however prove that such a BTM exists), it is necessary to master the subject. It is also true that the ultimate goal of most of the theory of BTMs is to provide efficient constructional techniques. With the idea that "bigger is better" we set out to construct a $BTM(19, 19; 6, 1, 8; 8, 3)$. Billington and Robinson [2] provide the necessary technique to construct such a BTM (which we call **Example 2.7**). However, it must be noted that although we will construct one BTM with the parameters $(19, 19; 6, 1, 8; 8, 3)$, there may very well be other (non-isomorphic) BTMs with the same parameters. Let the 19 elements in the BTM be $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18\} = Z_{19}$. We can construct the 19 blocks by starting with the initial block $b_1 = [0, 0, 1, 4, 6, 7, 9, 11]$. To obtain block b_i from block $b_{(i-1)}$ where $i \leq 19$ we add one to each of b_i and change any instances of 19 to 0 (this is called addition modulo 19). So, for instance, we would have that $b_2 = [1, 1, 2, 5, 7, 8, 10, 12]$, $b_8 = [7, 7, 8, 11, 13, 14, 16, 18]$, and $b_9 = [8, 8, 9, 12, 14, 17, 0]$. The entire incidence matrix for this design becomes:

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 2 & 0 \end{bmatrix}$$

We have seen ([4]) that a binary error-correcting code can be formed using the following structure *:

$$\begin{bmatrix} \vec{0}_b \\ N \\ J_{vxb} - N \\ \vec{1}_b \end{bmatrix}$$

where N is the incidence matrix of a BIBD, and $J_{vxb} = (\vec{1}_v)^T \vec{1}_b$. Since, BTDs generalize BIBDs it makes sense to consider this same structure * used with the incidence matrix of a BTD rather than a BIBD. In this chapter we discuss some necessary conditions on ternary designs such that the incidence matrix of a ternary design gives a ternary error-correcting code under the structure *.

2.3. Results on Ternary Codes Through Ternary Designs

The first thing to consider is the possibility of two identical code words resulting from the structure *.

Theorem 2.8. *If two rows \vec{m}, \vec{n} from the incidence matrix N of a balanced ternary design satisfy $\vec{m} = \vec{1} - \vec{n}$, then the design's parameters are*

$$\left(\frac{\rho_2 + 2\rho_1}{\rho_1 - 4\rho_2}, 2\rho_1 + \rho_2; \rho_1, \rho_2, \rho_1 + 2\rho_2; \frac{\rho_1 + 2\rho_2}{\rho_1 - 4\rho_2}, 4\rho_2 \right).$$

Proof. Since $\vec{m} = \vec{1} - \vec{n}$ we have that, without loss of generality,

$$\begin{aligned}\vec{m} &= 22 \dots 211 \dots 100 \dots 0 \\ \vec{n} &= 22 \dots 200 \dots 011 \dots 1\end{aligned}$$

We conclude that $\Lambda = 4\rho_2$ since all the twos line up and none of the ones line up. We also have that $B = \rho_2 + 2\rho_1$ by counting the length of \vec{m} . The rest of the parameters follow from the design conditions $VR = BK$ and $\Lambda(V - 1) = \rho_1(K - 1) + 2\rho_2(K - 2)$. \square

Since we want to find out when $*$ gives a ternary error-correcting code, we need to investigate the minimum distance between code words given by $*$. We will have to find a lower bound on the minimum distance of any two code words given by $*$. Thus, we will sometimes need to assume the worst thing possible in relation to distance between codewords (the lowest possible distance). This way our result will encompass all BTDs, and their behavior in $*$.

Lemma 2.9. *The distance between the two rows $\vec{0}$ and \vec{r} , where \vec{r} is a row from the incidence matrix of a BTD, is $\rho_1 + \rho_2$.*

Proof. The number of differing positions between $\vec{0}$ and \vec{r} is the number of nonzero positions in \vec{r} . However \vec{r} is a row from the incidence matrix of a BTD, therefore \vec{r} has ρ_1 positions containing a 1 and ρ_2 positions containing a 2. All other positions in \vec{r} contains a zero. Hence we have that the number of nonzero positions in \vec{r} is $\rho_1 + \rho_2$. \square

Lemma 2.10. *The distance between two rows $\vec{0}$ and \vec{r}' where \vec{r}' is a row from $J_{vxb} - N$, where N is the incidence matrix of a BTD, is $B - \rho_1$.*

Proof. This distance is equal to the number of nonzero positions in \vec{r}' . However since \vec{r}' is a row in $J_{vxb} - N$ there exists a corresponding row \vec{r} in N such that $\vec{r}' = \vec{1} - \vec{r}$. Therefore all the positions in \vec{r}' that are $\vec{0}$, are the same as the positions in \vec{r} that are 1. Thus ρ_1 positions in \vec{r}' are zero and the rest are nonzero. However, the total number of positions is the blocksize B of the design. \square

Lemma 2.11. *The distance between two rows $\vec{1}$ and \vec{r} , where \vec{r} is from N , the incidence matrix of a BTD, is $B - \rho_1$.*

Proof. This distance is equal to the number of positions in \vec{r} that are not one. Summing the number of zeros and the number of twos in \vec{r} gives $B - \rho_1 - \rho_2 + \rho_2 = B - \rho_1$. \square

Lemma 2.12. *The distance between two rows $\vec{1}$ and \vec{r}' , where \vec{r}' is a row from $J_{vxb} - N$, where N is the incidence matrix of a BTD, is $\rho_1 + \rho_2$.*

Proof. Let \vec{r} be $\vec{1} - \vec{r}'$. The desired distance is equal to the number of positions in \vec{r} that are not zero. Summing the number of ones and the number of twos in \vec{r} gives $\rho_1 + \rho_2$. \square

Lemma 2.13. *The distance between two rows \vec{r}, \vec{r}' , where \vec{r} is from the incidence matrix of a BTD and $\vec{r}' = \vec{1} - \vec{r}$ is $B - \rho_2$.*

Proof. The rows \vec{r} and \vec{r}' will differ everywhere except where there are twos. Since \vec{r} is from the incidence matrix of a BTD it has ρ_2 twos, thus \vec{r} and \vec{r}' differ in $B - \rho_2$ places. \square

Theorem 2.14. *The distance between two distinct rows \vec{m}, \vec{n} from N , the incidence matrix of a BTD, is greater than $2(\rho_1 + \rho_2 - \Lambda)$ if $\Lambda \leq \rho_1$ or $2(\rho_2 - \frac{\Lambda - \rho_1}{4})$ if $\Lambda > \rho_1$.*

Proof. In any BTD the least possible distance between two rows happens when the number of positions where both rows have a '1' is maximized. In the following discussion we consider only this case, assuming least possible distance.

Suppose that $\Lambda \leq \rho_1$. Then, Λ ones line up in corresponding positions between \vec{m} and \vec{n} and these are the only places where nonzero entries line up. However there are still $\rho_1 - \Lambda$ ones and ρ_2 twos in both \vec{m} and \vec{n} . Thus the distance is $2(\rho_1 + \rho_2 - \Lambda)$.

Now, suppose that $\Lambda > \rho_1$. Now we have four cases to consider:

Case #1: $(\Lambda - \rho_1) \bmod 4 = 0$. In this case, all the ones line up in \vec{m} and \vec{n} and some twos line up. The number of twos that line up is $\frac{\Lambda - \rho_1}{4}$. This is the case because the inner product of \vec{m} and \vec{n} must be Λ . Without loss of generality, the structure is:

$$\begin{array}{cccccccc} 1 & \dots & 1 & 2 & 2 & \dots & 2 & 2 & \dots & 2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 1 & \dots & 1 & 2 & 2 & \dots & 2 & 0 & \dots & 0 & 2 & 2 & \dots & 2 & 2 & 0 & \dots & 0 \end{array}$$

So we have that the distance between the two is

$$2\left(\rho_2 - \frac{\Lambda - \rho_1}{4}\right) \tag{1}$$

Case #2 $(\Lambda - \rho_1) \bmod 4 = 1$. In this case $\rho_1 - 1$ ones line up between \vec{m} and \vec{n} , in one place a 1 lines up with a 2, and some 2s line up. This happens as a natural consequence of maximizing the number of 1s that line up. To see this notice that since $(\Lambda - \rho_1) \bmod 4 = 1$. If all the ones lined up, then the remaining part $\Lambda - \rho_1$ would not be divisible by 4. So even if as many twos lined up as possible, the inner product between \vec{m} and \vec{n} will still only be $\Lambda - 1$. But we need the inner product to be Λ so if one 1 lines up with a 2 and all the other ones line up then we can line up some twos to get Λ . Without loss of generality, the structure is:

$$\begin{array}{cccccccc} 1 & \dots & 1 & 1 & 0 & 2 & 2 & \dots & 2 & 2 & \dots & 2 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ 1 & \dots & 1 & 2 & 1 & 2 & 2 & \dots & 2 & 0 & \dots & 0 & 2 & 2 & \dots & 2 & 2 & 0 & \dots & 0 \end{array}$$

The distance is $2 + \rho_2 - \frac{\Lambda - \rho_1 - 1}{4} + \rho_2 - \frac{\Lambda - \rho_1 - 1}{4} - 1 = 1 + 2(\rho_2 - \frac{\Lambda - \rho_1 - 1}{4}) =$

$$\frac{3}{2} + 2(\rho_2 - \frac{\Lambda - \rho_1}{4}) \quad (2)$$

Case #3 $(\Lambda - \rho_1) \bmod 4 = 2$. In this case $\rho_1 - 2$ ones line up between \vec{m} and \vec{n} , and some twos line up. It may also be the case that 2 ones line up with 2 twos. It can be shown that this would give the same distance. We consider the former case where no one lines up with a two. Here we have that $\rho_1 - 2$ ones line up so 4 ones line up with zeros. Also $\frac{\Lambda - \rho_1 + 2}{4}$ twos must line up so $2(\rho_2 - \frac{\Lambda - \rho_1 - 2}{4} - 1)$ twos line up with zeros. Without loss of generality, the structure is:

$$\begin{aligned} 11 \dots 1110022 \dots 222 \dots 200 \dots 000 \dots 0 \\ 11 \dots 1001122 \dots 200 \dots 022 \dots 220 \dots 0 \end{aligned}$$

We have that the distance between \vec{m} and \vec{n} is $4 + 2(\rho_2 - \frac{\Lambda - \rho_1 - 2}{4} - 1) = 2 + 2(\rho_2 - \frac{\Lambda - \rho_1 - 2}{4}) =$

$$3 + 2(\rho_2 - \frac{\Lambda - \rho_1}{4}) \quad (3)$$

Case #4 $(\Lambda - \rho_1) \bmod 4 = 3$. In this case $\rho_1 - 1$ ones line up between m and n , and some twos line up. Hence we have that 2 ones line up with zeros. Also $\frac{\Lambda - \rho_1 + 1}{4}$ twos line up, so $2(\rho_2 - \frac{\Lambda - \rho_1 + 1}{4})$ twos line up with zeros. Without loss of generality, the structure is:

$$\begin{aligned} 11 \dots 11022 \dots 222 \dots 200 \dots 000 \dots 0 \\ 11 \dots 10122 \dots 200 \dots 022 \dots 220 \dots 0 \end{aligned}$$

We have that the distance between \vec{m} and \vec{n} is $2 + 2(\rho_2 - \frac{\Lambda - \rho_1 + 1}{4}) =$

$$\frac{3}{2} + 2(\rho_2 - \frac{\Lambda - \rho_1}{4}) \quad (4)$$

It is clear that the minimum of (1), (2), (3), and (4) is $2(\rho_2 - \frac{\Lambda - \rho_1}{4})$. \square

Corollary 2.15. *The distance between two distinct rows \vec{m}', \vec{n}' from $J_{vxb} - N$, where N is the incidence matrix of a BTD, is greater than $2(\rho_1 + \rho_2 - \Lambda)$ if $\Lambda \leq \rho_1$ or $2(\rho_2 - \frac{\Lambda - \rho_1}{4})$ if $\Lambda > \rho_1$.*

Proof. This is due to the fact that the distance between \vec{m}' and \vec{n}' is equal to the distance between \vec{m} and \vec{n} where $\vec{n} = \vec{1} - \vec{n}'$ and $\vec{m} = \vec{1} - \vec{m}'$. However, \vec{m} and \vec{n} are distinct rows from N , hence we have the desired result. \square

Theorem 2.16. *The distance between two distinct rows \vec{m}, \vec{n}' , where \vec{m} is from N , the incidence matrix of a BTD, and \vec{n}' is from $J_{vxb} - N$, is greater than $B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor$ if $\Lambda \leq 4\rho_2$ or $B + 2\Lambda - 9\rho_2 - 2\rho_1$ if $\Lambda \geq 4\rho_2$.*

Proof. Since \vec{n}' is from $J_{vxb} - N$ there is a corresponding row \vec{n} from N such that $\vec{n}' = \vec{1} - \vec{n}$. Intuitively, \vec{n}' is \vec{n} with the ones and zeros swapped. Lets examine the general structure of \vec{m} and \vec{n} lined up one on top of the other, with similar patterns grouped into blocks:

$$\begin{array}{cccccccccccccccc} 2 & 2 & \dots & 2 & | & 2 & 2 & \dots & 2 & | & 2 & 2 & \dots & 2 & | & 1 & 1 & \dots & 1 & | & 1 & 1 & \dots & 1 & | & 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ 2 & 2 & \dots & 2 & | & 1 & 1 & \dots & 1 & | & 0 & 0 & \dots & 0 & | & 2 & 2 & \dots & 2 & | & 1 & 1 & \dots & 1 & | & 0 & 0 & \dots & 0 & | & 2 & 2 & \dots & 2 & | & 1 & 1 & \dots & 1 & | & 0 & 0 & \dots & 0 \end{array}$$

Now we examine the corresponding general structure of \vec{m} and \vec{n}' :

$$\begin{array}{cccccccccccccccc} 2 & 2 & \dots & 2 & | & 2 & 2 & \dots & 2 & | & 1 & 1 & \dots & 1 & | & 1 & 1 & \dots & 1 & | & 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 & | & 0 & 0 & \dots & 0 \\ 2 & 2 & \dots & 2 & | & 0 & 0 & \dots & 0 & | & 1 & 1 & \dots & 1 & | & 2 & 2 & \dots & 2 & | & 0 & 0 & \dots & 0 & | & 1 & 1 & \dots & 1 & | & 2 & 2 & \dots & 2 & | & 0 & 0 & \dots & 0 & | & 1 & 1 & \dots & 1 \end{array}$$

Clearly the only blocks that don't add to the distance between the codewords \vec{m} and \vec{n}' are the first, the sixth, and the eighth. If we are looking for a lower bound on distance, we would want to maximize the size of these blocks, that is look for the worst case (when the size of these blocks are biggest). This is equivalent to maximizing the size of blocks 1, 6, and 8 among the former structure comparison between \vec{m} and \vec{n} . Clearly to do this, we want to line up as many corresponding twos as possible, this directly maximizes block 1, however it eliminates the need to line up a lot of ones, so lots of ones will line up with zeros, indirectly maximizing blocks 6 and 8. We must think about the case where lining up twos does not suffice to produce Λ . This could occur if all the twos are lined up but Λ is not achieved, or if Λ is not divisible by 4 so something other than twos need to be lined up. The latter case is very common. We will always line up as many twos as possible. If we use all the twos up doing so(former case), then we simply line up as many ones that are left. However if we have enough twos but the problem is that Λ is not divisible by 4, then we still line up as many twos as possible but we still may have 1, 2, or 3 left over to reach Λ . If it is 1 we have no choice but to line up one pair of ones. Otherwise we may line up 1 one with a two, or line up several ones. It is best to line up 1 one with a two, because this indirectly increases the size of blocks 6 and 8 in comparison to only lining up ones which decreases the size of blocks 6 and 8.

Case #1 $\Lambda \leq 4\rho_2$: As discussed we want to line up as many twos as possible. We can see that we want to line up $\lfloor \frac{\Lambda}{4} \rfloor$ twos. Now if $\Lambda \bmod 4$ is 2 or 3 we want to line up a two with a one, so in general we want to line up $\lfloor \frac{\Lambda \bmod 4}{2} \rfloor$ twos from m with ones from n . This number will be either 0 or 1, and corresponds to block 2 from the general structure above. So we have that there will be $\rho_2 - \lfloor \frac{\Lambda}{4} \rfloor - \lfloor \frac{\Lambda \bmod 4}{2} \rfloor$ twos left in \vec{m} that will need to be lined up with zeros from \vec{n} (block 3) and $\rho_2 - \lfloor \frac{\Lambda}{4} \rfloor$ twos from \vec{n} that will need to be lined up with zeros from \vec{m} (block 7). However Λ still may not be achieved. Hence we need to line up $\Lambda - 4\lfloor \frac{\Lambda}{4} \rfloor - 2\lfloor \frac{\Lambda \bmod 4}{2} \rfloor$ ones between \vec{m} and \vec{n} . This number will again be either zero or one and corresponds to block 5 from the general structure above. We don't line up any ones from \vec{m} with twos from \vec{n} , thus block 4 from above is nonexistent. We can see from the general structure of \vec{m} and \vec{n}' that the only blocks adding distance to these codewords are 2,3,4,5,7, and 9. We have found the size of blocks 2,3,4,5,and 7, but what about 9. The size of block 9 is just the block size B minus the size of all the other blocks. To find this we need to consider the size of blocks 6 and 8. The size of block 6 equals the

number of ones from \vec{m} lined up with zeros from \vec{n} . But ones from \vec{m} are either lined up with ones from \vec{n} or zeros from \vec{n} and we know how many are lined up with ones from \vec{n} , hence the size of block 6 is $\rho_1 - \Lambda + 4\lfloor \frac{\Lambda}{4} \rfloor + 2\lfloor \frac{\Lambda \bmod 4}{2} \rfloor$. The size of block 8 equals the number of ones from \vec{n} lined up with zeros from \vec{m} . Ones from \vec{m} may be lined up with either zeros, ones, or twos from \vec{n} but we already know how many are lined up with ones and twos. Hence the size of block 8 is $\rho_1 - \Lambda + 4\lfloor \frac{\Lambda}{4} \rfloor + \lfloor \frac{\Lambda \bmod 4}{2} \rfloor$. Finding the distance is now simplified to adding the size of blocks 2,3,4,5,7, and 9, which gives the distance

$$B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor$$

Case #2 $\Lambda \geq 4\rho_2$: This case is simpler because we just line up all the twos from \vec{m} and \vec{n} and line up as many ones from \vec{m} and \vec{n} that are needed. We need to line up ρ_2 twos (block 1), $\Lambda - 4\rho_2$ ones (block 5), $2(\rho_1 - \Lambda + 4\rho_2)$ ones with zeros (blocks 6 and 8 combined), and $B - 5\rho_2 - 2\rho_1 + \Lambda$ zeros (block 9). All other blocks are nonexistent in this case. Adding the size of blocks 2,3,4,5,7, and 9 gives the distance

$$B + 2\Lambda - 9\rho_2 - 2\rho_1$$

This concludes case #2. \square

Conclusion 2.17. *If N is the incidence matrix of a BTD satisfying $\Lambda \leq \rho_1$ and $\Lambda \leq 4\rho_2$ then N under the structure $*$ gives a one error-correcting ternary code if the every element of the set $\{ \rho_1 + \rho_2, B - \rho_1, B - \rho_2, 2(\rho_1 + \rho_2 - \Lambda), B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor \}$ is greater than or equal to 3.*

Conclusion 2.18. *If N is the incidence matrix of a BTD satisfying $\Lambda > \rho_1$ and $\Lambda \leq 4\rho_2$ then N under the structure $*$ gives a one error-correcting ternary code if the every element of the set $\{ \rho_1 + \rho_2, B - \rho_1, B - \rho_2, 2(\rho_2 - \frac{\Lambda - \rho_1 + 1}{4}), B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor \}$ is greater than or equal to 3.*

Conclusion 2.19. *If N is the incidence matrix of a BTD satisfying $\Lambda \leq \rho_1$ and $\Lambda \geq 4\rho_2$ then N under the structure $*$ gives a one error-correcting ternary code if the every element of the set $\{ \rho_1 + \rho_2, B - \rho_1, B - \rho_2, 2(\rho_1 + \rho_2 - \Lambda), B + 2\Lambda - 9\rho_2 - 2\rho_1 \}$ is greater than or equal to 3.*

Conclusion 2.20. *If N is the incidence matrix of a BTD satisfying $\Lambda > \rho_1$ and $\Lambda \geq 4\rho_2$ then N under the structure $*$ gives a one error-correcting ternary code if the every element of the set $\{ \rho_1 + \rho_2, B - \rho_1, B - \rho_2, 2(\rho_2 - \frac{\Lambda - \rho_1 + 1}{4}), B + 2\Lambda - 9\rho_2 - 2\rho_1 \}$ is greater than or equal to 3.*

2.4. Examples

Example 2.21. In Example 2.7 we have constructed a $BTD(19, 19; 6, 1, 8; 8, 3)$. Conclusion 2.17. tells us that this produces a one error-correction code (corrects at least 1 error, this one corrects more). The code (for brevity we omit the parentheses and commas) that is constructed is:

```
{
    00000000000000000000, 2000000010101101001, 1200000001010110100,
    0120000000101011010, 0012000000010101101, 1001200000001010110,
    0100120000000101011, 1010012000000010101, 1101001200000001010,
    0110100120000000101, 1011010012000000010, 0101101001200000001,
    1010110100120000000, 0101011010012000000, 0010101101001200000,
    0001010110100120000, 0000101011010012000, 0000010101101001200,
    0000001010110100120, 0000000101011010012, 2111111101010010110,
    0211111110101001011, 1021111111010100101, 1102111111101010010,
    0110211111110101001, 1011021111111010100, 0101102111111101010,
    0010110211111110101, 1001011021111111010, 0100101102111111101,
    1010010110211111110, 0101001011021111111, 1010100101102111111,
    1101010010110211111, 1110101001011021111, 1111010100101102111,
    1111101010010110211, 1111110101001011021, 1111111010100101102,
    11111111111111111111
}
```

If we look closely at the results used to justify Conclusion 2.17 we see that the minimum distance of this code is guaranteed to be greater than the minimum of $\{ \rho_1 + \rho_2, B - \rho_1, B - \rho_2, 2(\rho_1 + \rho_2 - \Lambda), B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor \} = \{7, 13, 18, 8, 10\}$. We know from Lemmas 2.9, 2.10, 2.11, 2.12, and 2.13 that there will be codewords with minimum distance 7, 13, and 18 (so this code is really a three error-correcting code), however Theorem 2.14 states that the distance between two codewords both from the incidence matrix N is greater than or equal to 8. Also, Theorem 2.16 ensures the distance between codewords \vec{m} and \vec{n}' where \vec{m} is from N and \vec{n}' is from $J_{vxb} - N$ is greater than or equal to 10. Since these are lower bounds it is natural to ask how good are these bounds. Is equality every achieved? This example shows that in both of these bounds equality can be achieved. The two codewords

```
2000000010101101001
0012000000010101101
```

have distance 8 and equality is satisfied in the first bound and

```
0012000000010101101
0100101102111111101
```

are codewords of distance 10 that prove equality can occur in the second bound. In both cases, just picking two codewords from the specified place will not guarantee equality because, for example

2000000010101101001
1010012000000010101

are two codewords from N with distance $11 > 8$, while

2000000010101101001
1111110101001011021

are two codewords with distance $14 > 10$ where one is from N and the other from $J_{vxb} - N$.

Example 2.22. Using Billington and Robinson [2], we can construct a

$BTD(4, 8; 2, 3, 8; 4, 6)$. Conclusion 2.18 tells us that this produces a one error-correcting code. It satisfies the conditions: $\rho_1 + \rho_2 = 2 + 3 = 5 \geq 3$, $B - \rho_1 = 8 - 2 = 6 \geq 3$, $B - \rho_2 = 8 - 3 = 5 \geq 3$, $2(\rho_2 - \frac{\Lambda - \rho_1 + 1}{4}) = 2(3 - \frac{5}{4}) = 2(\frac{7}{4}) = \frac{7}{2} \geq 3$, $B - 9\lfloor \frac{\Lambda}{4} \rfloor - 2\rho_1 + 2\Lambda - 3\lfloor \frac{\Lambda \bmod 4}{2} \rfloor = 8 - 9(1) - (2)(2) + (2)(6) - (3)(1) = 4 \geq 3$, thus it is a one error-correcting. The code that is constructed is:

{
 00000000, 11222000, 11200220, 11020202, 11002022,
 00222111, 00211221, 00121212, 00112122, 11111111
}

Example 2.23. Using Billington and Robinson [2], we can construct a BTD with the parameters $(9, 18; 10, 2, 14, 7, 10)$ Conclusion 2.19 does not help because the lower bound on the distance is less than zero. However, the code is at least a one error-correcting code:

{
 000000000000000000, 210110011201101011, 121011001120110101,
 112101100112011010, 011210110011201101, 001121011101120110,
 100112101010112011, 110011210101011201, 011001121110101120,
 101100112011010112, 201001100210010100, 020100110021001010,
 002010011002100101, 100201001100210010, 110020100010021001,
 011002010101002100, 001100201010100210, 100110020001010021,
 010011002100101002, 111111111111111111
}

The lower bound given by Conclusion 2.19 is not ideal in this example because the assumption that as many 2s as possible will line up. In this example no pairs of 2s line up so we actually get good distances.

Example 2.24. Using Billington and Robinson [2], we can construct a BTD with the parameters $(3, 11; 7, 2, 11; 3, 9)$. Conclusion 2.20 does not help because the lower bound on the distance is less than zero. However when we construct the code we do get a one error-correcting code:

{
 00000000000, 20120111111, 12012011111, 01201211111
 21021000000, 02102100000, 10210200000, 11111111111
}

The minimum distance of the code is 4 so it is one error-correcting. Why does Conclusion 2.20 have a lower bound less than zero? It has to do with the case of the distance between a row \vec{m} of the incidence matrix N and a row \vec{n}' from $J_{vxb} - N$. Then it assumes that all the twos line up between \vec{m} and $\vec{n} = \vec{1} - \vec{n}'$. If this happened in this design then two pairs of 2s would line up and one pair of 1s would line up. Then the distance between \vec{m} and \vec{n}' would be negative. However, this is clearly impossible since more than one pair of 1s must line up. Hence two 2s cannot possibly line up. This suggests that some additional results might be obtained under the conditons of Conclusion 2.20 where the additional condition of $\rho_1 - \Lambda + 4\rho_2 > B - \rho_1 - \rho_2$ is satisfied. However, under these conditions it becomes very complicated to describe the distances of codewords for general BTDs. Since the minimum distance of the code is 4, no more than 1 error can be corrected.

The four preceding examples illustrate each of the four conclusions. It is clear that while Conclusion 2.17 and Conclusion 2.18 give very good lower bounds, Conclusion 2.19 and Conclusion 2.20 give lower bounds that are sometimes not very helpful. These Conclusions (along with their associated Theorems) are useful from a theoretical viewpoint since they provide at the very least a starting point for the theory under those conditions and they do provide lower bounds. However, for the practitioner only interested in creating Ternary codes, the lower bounds of Conclusions 2.19 and 2.20 associated with Theorem 2.16 (and in some cases Theorem 2.14) should not discourage the use of certain designs.

Chapter 3. Linear Codes Through Latin Squares

In this chapter we explore one possible relationship between Latin Squares and error-correcting codes (especially error-correction codes).

3.1. Introduction and Basic Definitions

Definition. A matrix M is a **latin square of order n** (or an **$n \times n$ latin square**) if its columns and rows are permutations of n fixed elements.

A permutation of n symbols is just an arrangement of those symbols. For example $abcd$ and $badc$ are permutations of the four symbols a, b, c, d . Basically, a latin square is a matrix with n rows and n columns (thus a square matrix), such that only n elements appear in the matrix and no element appears twice in any row or column. Here is a simple example of a latin square:

$$\begin{bmatrix} a & b & c & d \\ c & a & d & b \\ d & c & b & a \\ b & d & a & c \end{bmatrix}$$

Since there are exactly n elements in any given latin square of order n and no element repeats itself in a row or column, each column contains all n elements.

Notation. In the following discussion the rows and columns of an $n \times n$ matrix are indexed by $0, 1, 2, \dots, n - 1$. All elements in the latin squares in this paper are from Z_n , where addition and multiplication are defined modulo n . Therefore, all operations performed on these elements, in this paper, are assumed to be performed modulo n , however we often state this explicitly for emphasis. Finally, for all latin squares of order n we assume $n > 1$.

Definition. Two $n \times n$ latin squares $A = \|a_{ij}\|$ and $B = \|b_{ij}\|$ are **orthogonal** if $|\{(a_{ij}, b_{ij}) \mid i, j \in \{0, 1, 2, \dots, n - 1\}\}| = n^2$. A set of $t > 0$ latin squares are **pairwise mutually orthogonal** (or just **mutually orthogonal**) if every pair of latin squares in the set is orthogonal.

Another way to define the concept of orthogonal latin squares is to say that two latin squares are orthogonal if no pair of elements made from corresponding entries in the two latin squares is repeated. Since none of the pairs are repeated and there are n^2 total pairs (since there are n^2 entries in the latin squares) all of the pairs appear.

Definition. A code C is **linear** if the addition of any two codewords is another codeword.

Latin squares are dealt with extensively in Denes and Keedwell [1]. In particular, they showed a method of constructing an error-correcting code of distance $t + 1$ with n^2 codewords of length $t + 2$ when given t mutually orthogonal latin squares. Their method is as follows:

Given t mutually orthogonal latin squares L_1, L_2, \dots, L_t , the code is the set of all codewords of the form $(i, j, l_1, l_2, \dots, l_t)$ where l_1 is the i, j -th entry of L_1 , l_2 is the i, j -th entry of L_2 , and l_k is the i, j -th entry of L_k where $1 \leq k \leq t$.

This method of construction was also rediscovered by Kadowaki, Kageyama, Kimura, and Yanagida [2]. In particular, they gave the following example of a perfect code (every vector is correctable) using two orthogonal latin squares of order 3. With our notation the two latin squares are:

$$\begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \\ 2 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \\ 1 & 2 & 0 \end{bmatrix}$$

The code constructed using these two is

$$\{(0, 0, 0, 0), (0, 1, 1, 1), (0, 2, 2, 2), (1, 0, 1, 2), (1, 1, 2, 0), \\ (1, 2, 0, 1), (2, 0, 2, 1), (2, 1, 0, 2), (2, 2, 1, 0)\}$$

A noteworthy feature of this code is that it is also a linear code when addition and multiplication are defined modulo n . The question that arises naturally is when, if ever, do other linear codes result from this construction? When a code C results from applying the above method to some set of mutually orthogonal latin squares we say that these latin squares **generate** C . If C is a linear code we say that these latin squares **generate a linear code modulo n** , where n is the order of the latin squares.

In this chapter we give the general structure of all latin squares that generate linear codes modulo n . Furthermore, we provide necessary and sufficient conditions for two such latin squares to be orthogonal. This completely characterizes all sets of mutually orthogonal latin squares of order n that generate linear codes modulo n . We also provide an upper bound on the number of mutually orthogonal latin squares that generate linear codes modulo n . Lastly, we show by construction that this upper bound is attainable.

3.2. Results

Theorem 2.1. *If an $n \times n$ latin square $L = \| l_{ij} \|$ generates a linear code modulo n then $l_{00} = 0$.*

Proof. Since L generates a linear code modulo n the addition of any two codewords is another codeword. Thus $2(0, 0, l_{00}) = (0, 0, 2l_{00})$ is a codeword. However, this implies that $l_{00} = 2l_{00}$, which gives $l_{00} = 0$. \square

Lemma 2.2. *An $n \times n$ matrix $L = \| l_{ij} \|$ of the form $l_{ij} = (i\beta + j\alpha) \bmod n$ for some integers α, β in the range $0 < \alpha, \beta < n$ is a latin square iff $\gcd(\alpha, n) = \gcd(\beta, n) = 1$.*

Proof. First we show that if $\{\gcd(\alpha, n), \gcd(\beta, n)\} \neq \{1\}$ then $L = \| l_{ij} \|$ where $l_{ij} = (i\beta + j\alpha) \bmod n$ is not a latin square. Assume $\gcd(\alpha, n) > 1$, then for some integer k , $0 < k < n$ we have that $k\alpha \bmod n = 0$. This is true because the elements of L , $\{0, 1, \dots, n-1\}$ form a cyclic group G with addition modulo n . One generator of this group is 1. From Fraleigh [4], we have that the cyclic subgroup of G generated by α has order $\frac{n}{\gcd(\alpha, n)}$. Thus, if $\gcd(\alpha, n) > 1$, α generates a cyclic subgroup that doesn't contain all n elements of G and hence for some integer k , $0 < k < n$, $k\alpha \bmod n = 0$. But this means that $l_{00} = l_{0k} = 0$ and L has two zeros in its first row. Hence L cannot be a latin square. Similarly, if $\gcd(\beta, n) > 1$, there exists some integer k , $0 < k < n$, such that $k\beta \bmod n = 0$, but then $l_{00} = l_{k0} = 0$ and L has two zeros in its first column and therefore cannot be a latin square. We have shown by contrapositive that if $L = \| l_{ij} \|$ is a latin square of the form $l_{ij} = (i\beta + j\alpha) \bmod n$ for some integers α, β in the range $0 < \alpha, \beta < n$, then $\gcd(\alpha, n) = \gcd(\beta, n) = 1$. Next, assume that $\gcd(\alpha, n) = \gcd(\beta, n) = 1$, and suppose $l_{ij_1} = l_{ij_2}$ for some integers i, j_1, j_2 in the range $0 < i, j_1, j_2 < n$. This implies that $i\beta + j_1\alpha \equiv i\beta + j_2\alpha \pmod{n}$ which gives $(j_1 - j_2)\alpha \equiv 0 \pmod{n}$. However, since $\gcd(\alpha, n) = 1$, $j_1 = j_2$. Similarly if $l_{i_1j} = l_{i_2j}$ for some $0 < i_1, i_2, j < n$ we have that $i_1 = i_2$ since $\gcd(\beta, n) = 1$. This means that L has no row or column with repeated elements, and thus L is a latin square. \square

Lemma 2.3. *An $n \times n$ latin square $L = \| l_{ij} \|$ of the form $l_{ij} = (i\beta + j\alpha) \bmod n$ for some integers α, β in the range $0 < \alpha, \beta < n$ generates a linear code modulo n .*

Proof. Let C be the code generated from L . Consider the two codewords $c_1 = (i_1, j_1, i_1\beta + j_1\alpha)$ and $c_2 = (i_2, j_2, i_2\beta + j_2\alpha)$. Then

$$c_1 + c_2 = (z_1, z_2, (z_1\beta + z_2\alpha) \bmod n)$$

where $z_1 = (i_1 + i_2) \bmod n$ and $z_2 = (j_1 + j_2) \bmod n$. Thus, from the structure of L we have that $c_1 + c_2 \in C$. \square

Lemma 2.4. *If $L = \| l_{ij} \|$ is an $n \times n$ latin square that generates a linear code modulo n , then L is of the form $l_{ij} = (i\beta + j\alpha) \pmod n$ for some integers α, β in the range $0 < \alpha, \beta < n$ and $\gcd(\alpha, n) = \gcd(\beta, n) = 1$.*

Proof. From Theorem 2.1, $l_{00} = 0$. Let $l_{01} = \alpha$ and $l_{10} = \beta$ for some integers α, β in the range $0 < \alpha, \beta < n$. Now let C be the code generated from L . Assume C is a linear code modulo n . Then any multiple of a codeword is a codeword. Thus, for any integer k , $0 \leq k < n$, $k(0, 1, \alpha) = (0, k, k\alpha)$ and we have that $l_{0k} = k\alpha$. Also $k(1, 0, \beta) = (k, 0, k\beta)$ and we have that $l_{k0} = k\beta$. Furthermore, for any $0 < i, j < n$, we have that

$$i(1, 0, \beta) + j(0, 1, \alpha) = (i, 0, i\beta) + (0, j, j\alpha) = (i, j, i\beta + j\alpha)$$

Hence, $l_{ij} = i\beta + j\alpha$. From Lemma 2.2, for L to be a latin square, $\gcd(\alpha, n) = \gcd(\beta, n) = 1$. \square

Theorem 2.5. *An $n \times n$ matrix $L = \| l_{ij} \|$ is a latin square that generates a linear code modulo n iff L is of the form $l_{ij} = (i\beta + j\alpha) \pmod n$ for some integers α, β satisfying 1) $0 < \alpha, \beta < n$ and 2) $\gcd(\alpha, n) = \gcd(\beta, n) = 1$.*

Proof. Follows immediately from Lemmas 2.2, 2.3, and 2.4. \square

The above theorem characterizes every latin square that is a possible member of a set of mutually orthogonal latin squares that generate a linear code modulo n . The theorem ensures that the structure of these latin squares will be:

$$\begin{bmatrix} 0 & \alpha & 2\alpha & \dots \\ \beta & \beta + \alpha & \beta + 2\alpha & \dots \\ 2\beta & 2\beta + \alpha & 2\beta + 2\alpha & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix}$$

Latin squares of even order are not very useful in terms of generating linear codes modulo n as the following result shows.

Theorem 2.6. *If n is an even positive integer, then there is no pair of $n \times n$ mutually orthogonal latin squares that generate a linear code modulo n .*

Proof. Let $A = \| a_{ij} \|$ and $B = \| b_{ij} \|$ be two $n \times n$ mutually orthogonal latin squares that generate a linear code with $n = 2k$ for some positive integer k . Then, by Theorem 2.1, $2(0, k, a_{0k}, b_{0k}) = (0, 2k, 2a_{0k}, 2b_{0k}) = (0, 0, 2a_{0k}, 2b_{0k}) = (0, 0, 0, 0)$. This means that $2a_{0k} = 0$ and $2b_{0k} = 0$. We have that $a_{0k} \neq 0$ and $b_{0k} \neq 0$ because 0 already occurs in the first rows of A and B . Thus, we clearly have that $a_{0k} = b_{0k} = k$. However, we also have $2(k, 0, a_{k0}, b_{k0}) = (0, 0, 2a_{k0}, 2b_{k0})$, hence $a_{k0} = b_{k0} = k$. Therefore

$$(a_{0k}, b_{0k}) = (a_{k0}, b_{k0}) = (k, k)$$

and we have that A and B are not orthogonal, a contradiction. \square

Notation. If $A = (\| a_{ij} \|)$ is a latin square that generates a linear code and is of the form $a_{ij} = (i\beta + j\alpha) \bmod n$ for some integers α, β in the range $0 < \alpha, \beta < n$, then we denote A by $A = (\| a_{ij} \|, \alpha, \beta)$.

This notation emphasizes the fact that any $n \times n$ latin square used to generate a linear code is completely described by its α and β .

Lemma 2.7. Let $A = (\| a_{ij} \|, \alpha, \beta)$ and let g be some integer in the range $0 \leq g < n$. Then, g occurs in the i -th row of A at the position $a_{i, g\alpha^{-1} - i\beta\alpha^{-1}}$.

Proof. We know that g occurs in the i -th row of A because A is a latin square. Thus, for some integer $k_i, 0 \leq k_i < n$, we have that $i\beta + k_i\alpha = g$ which implies that $k_i = (g - i\beta)\alpha^{-1} = g\alpha^{-1} - i\beta\alpha^{-1}$ \square

Let L be a latin square of order n of the form $L = (\| l_{ij} \|, \alpha, \beta)$. Now, since L is a latin square no symbol can appear twice in the same column of L . Therefore, suppose we fix some symbol, say g (and we know that g will be an integer in the range $0 \leq g < n$). Thus, the question arises as to whether the position of g in consecutive rows (or columns) is somehow related. We know from Lemma 2.7 that the position of g in the i th row is $g\alpha^{-1} - i\beta\alpha^{-1}$, and similarly the position of g in the $(i - 1)$ th row is $g\alpha^{-1} - (i - 1)\beta\alpha^{-1}$. So the change in position is

$$(g\alpha^{-1} - i\beta\alpha^{-1}) - (g\alpha^{-1} - (i - 1)\beta\alpha^{-1}) = -\beta\alpha^{-1}$$

which is independent of the value of g . The column analogue of Lemma 2.7 is that g occurs in the j -th column of L at the position $l_{g\beta^{-1} - j\alpha\beta^{-1}, j}$. Hence, considering the j th and $(j - 1)$ th column we have that the change in position between consecutive columns of the element g is

$$(g\beta^{-1} - j\alpha\beta^{-1}) - (g\beta^{-1} - (j - 1)\alpha\beta^{-1}) = -\alpha\beta^{-1}$$

and is again independent of the value of g . These observations give a lot of insight into the types of latin squares considered. For one thing, this tells us that when the main diagonal of L consists of the same element repeated n times, we must have $-\beta\alpha^{-1} = -\alpha\beta^{-1} = 1$, and this indeed points to the general fact that $(-\beta\alpha^{-1})^{-1} = -\alpha\beta^{-1}$ (certainly it is the case that $1^{-1} = 1$). So, we conclude that if $-\beta\alpha^{-1} = -\alpha\beta^{-1}$ then the upper triangular part of L is in some sense the same as the lower triangular part of L . More formally, this condition implies that if $g = l_{i_1 j_1}$, $k = l_{j_1 i_1}$, and $g = l_{i_2 j_2}$ then it will necessary follow that $k = l_{j_2 i_2}$. In the specific instance when $\alpha = \beta$ this is trivially true since the latin square will be symmetric. Now suppose that we are told that $-\beta\alpha^{-1} = -\alpha\beta^{-1}$, so each element

g has a corresponding element k with the properties listed above. The question that now arises, is how do we find out what k is? First we note that if $-\beta\alpha^{-1} = -\alpha\beta^{-1}$ then

$$\alpha\beta^{-1} + \beta\beta^{-1} = (\alpha + \beta)\beta^{-1}$$

and

$$\beta\alpha^{-1} + \alpha\alpha^{-1} = (\alpha + \beta)\alpha^{-1}$$

Therefore combining these two yields $(\alpha + \beta)\beta^{-1} = (\alpha + \beta)\alpha^{-1}$. This tells us that $(\alpha + \beta)$ cannot be relatively prime with n unless $\alpha = \beta$. For example, consider the following latin square that satisfies the condition in question:

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

Here we can see that $\alpha + \beta = 4$, which is not relatively prime with $n = 8$. When n is prime, we have shown that $\alpha = n - \beta$. In general, using the fact that g appears in row 0 of L at the position $g\alpha^{-1}$, we have that the element that appears in the first column at the same position is $g\beta\alpha^{-1}$ (let's call this g 's **partner**), so that is the answer to our original question. However, when n is prime we have that the sum of g and its partner is 0 because $g + g\beta\alpha^{-1} = g(1 + \beta\alpha^{-1}) = g(\alpha\alpha^{-1} + \beta\alpha^{-1}) = g(\alpha + \beta)\alpha^{-1} = 0$, since $\alpha + \beta = 0$. Using the same argument it can be seen that in general we have that the sum of g and its partner is never relatively prime with n . Here's another example:

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

In this example, $n = 7$, $\alpha = 2$, and $\beta = 5$. The partners of 0, 1, 2, 3, 4, 5, 6 are 0, 6, 5, 4, 3, 2, 1, respectively. Notice that since n is prime, we have that the sum of each of the elements with its partner is equal to $7 = n$, unless the element is 0.

The following theorem provides necessary and sufficient conditions for two latin squares that generate linear codes modulo n by themselves to be orthogonal. Two such orthogonal latin squares when taken together generate another linear code modulo n .

Theorem 2.8. Let $A = (\| a_{ij} \|, \alpha_1, \beta_1)$ and $B = (\| b_{ij} \|, \alpha_2, \beta_2)$. Then, A and B are orthogonal iff $\gcd((\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) = 1$.

Proof. Assume that $\gcd((\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) = 1$. Now assume that two corresponding entries of A and B are equal: $(g, h) = (a_{i_1j_1}, b_{i_1j_1}) = (a_{i_2j_2}, b_{i_2j_2})$. Then, from Lemma 2.7, we have

$$j_1 = g\alpha_1^{-1} - i_1\beta_1\alpha_1^{-1} = h\alpha_2^{-1} - i_1\beta_2\alpha_2^{-1} = j_1 \quad (1)$$

$$j_2 = g\alpha_1^{-1} - i_2\beta_1\alpha_1^{-1} = h\alpha_2^{-1} - i_2\beta_2\alpha_2^{-1} = j_2 \quad (2)$$

Subtracting (1) from (2) yields

$$\begin{aligned} i_1\beta_1\alpha_1^{-1} - i_2\beta_1\alpha_1^{-1} &= i_1\beta_2\alpha_2^{-1} - i_2\beta_2\alpha_2^{-1} \\ \Rightarrow i_1\beta_1\alpha_1^{-1} - i_2\beta_1\alpha_1^{-1} - i_1\beta_2\alpha_2^{-1} + i_2\beta_2\alpha_2^{-1} &= 0 \\ \Rightarrow i_1(\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}) - i_2(\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}) &= 0 \\ \Rightarrow (i_1 - i_2)(\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}) &= 0. \end{aligned}$$

We have that $i_1 = i_2$ since $\gcd((\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) = 1$. Comparing (1) and (2) we see that $j_1 = j_2$.

Now, assume $\gcd((\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) > 1$, then for some integer k , $0 < k < n$ we have that $k(\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}) = 0$. From Lemma 2.7, 0 occurs in the k -th row in A at $-k\beta_1\alpha_1^{-1}$ and in B at $-k\beta_2\alpha_2^{-1}$ but

$$\begin{aligned} k(\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}) = 0 &\Rightarrow k\beta_2\alpha_2^{-1} = k\beta_1\alpha_1^{-1} \\ &\Rightarrow -k\beta_2\alpha_2^{-1} = -k\beta_1\alpha_1^{-1} \end{aligned}$$

This means that the pair $(0, 0)$ occurs twice among corresponding entries from A and B and thus A and B are not orthogonal. \square

Corollary 2.9. Let $A = (\| a_{ij} \|, \alpha_1, \beta_1)$ and $B = (\| b_{ij} \|, \alpha_2, \beta_2)$. Then

- (1) If $\alpha_1 = \beta_1$ then A and B are orthogonal only if $\alpha_2 \neq \beta_2$.
- (2) If $\alpha_1 = \beta_1$ then A and B are orthogonal iff $\gcd(\alpha_2 - \beta_2, n) = 1$.
- (3) If $\alpha_1 = \alpha_2$ then A and B are orthogonal iff $\gcd(\beta_2 - \beta_1, n) = 1$.
- (4) If $\alpha_1 = \beta_1 = \alpha_2 \neq \beta_2$ then A and B are orthogonal iff $\gcd(\beta_2 - \alpha_1, n) = 1$.

Proof. We prove (2), and the rest follow obviously. Assume $\alpha_1 = \beta_1$. Applying Theorem 2.8, we have that A and B are orthogonal iff

$$\begin{aligned} 1 &= \gcd((\beta_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) = \gcd((\alpha_1\alpha_1^{-1} - \beta_2\alpha_2^{-1}), n) \\ &= \gcd((1 - \beta_2\alpha_2^{-1}), n) = \gcd((\alpha_2 - \beta_2), n) = 1 \end{aligned}$$

Note that we used the fact that $\gcd(\alpha_2, n) = \gcd(\alpha_2^{-1}, n) = 1$ in the last step above. (1) and (4) follow directly from (2), and (3) follows from Theorem 2.8. \square

It is of interest to know how many mutually orthogonal latin squares of some order n exist that together generate a linear code modulo n . One way to go about solving this

problem is to write a computer program that examines all sets of latin squares (of this type) of some fixed order and returns the size of the largest set of mutually orthogonal latin squares it could find. Such a computer program (written in LISP) is included as Appendix A of this essay. Here is a table with output from the program.

| n | k |
|-----|-----|
| 2 | 1 |
| 3 | 2 |
| 4 | 1 |
| 5 | 4 |
| 9 | 2 |
| 15 | 2 |
| 21 | 2 |

In this table n is the order of the latin squares and k is the order of a maximal set of mutually orthogonal latin squares of order n that generates a linear code modulo n . Of course, just to provide this table the program took several hours of execution on a standard PC. However, the program does provide enough data to detect the appropriate pattern, which is that k appears to be one less than the smallest prime factor of n . This leads to the following theorem which gives the upper bound. After that we provide a construction that achieves this upper bound for any n . This construction can be used to generate a linear code modulo n with maximum error-correction from a set of mutually orthogonal latin squares.

Theorem 2.10. *Suppose that the prime factorization of n is $n = p_1 p_2 \dots p_h$ such that $p_1 \leq p_2 \leq \dots \leq p_h$ and p_1, p_2, \dots, p_h are prime. Then there are at most $p_1 - 1$ mutually orthogonal latin squares of order n that generate a linear code modulo n .*

Proof. Suppose that there exists a set S of more than $p_1 - 1$ mutually orthogonal latin squares of order n that generate a linear code modulo n . Fix one of the latin squares in S , say $A = (\| a_{ij} \|, \alpha_1, \beta_1)$. Consider the set of differences:

$$D = \{(\beta_1 \alpha_1^{-1} - \beta_m \alpha_m^{-1}) \mid (\| l^m_{ij} \|, \alpha_m, \beta_m) \in (S - \{A\})\} \text{ mod } p_1$$

Suppose that there exist two latin squares $B = (\| b_{ij} \|, \alpha_2, \beta_2)$ and $C = (\| c_{ij} \|, \alpha_3, \beta_3)$ in $S - \{A\}$ such that $\beta_1 \alpha_1^{-1} - \beta_2 \alpha_2^{-1} \equiv \beta_1 \alpha_1^{-1} - \beta_3 \alpha_3^{-1} \pmod{p_1}$. This implies that $\beta_2 \alpha_2^{-1} - \beta_3 \alpha_3^{-1} \equiv 0 \pmod{p_1}$. However, by Theorem 2.8 we have that B and C are not orthogonal because $\gcd(\beta_2 \alpha_2^{-1} - \beta_3 \alpha_3^{-1}, n) \neq 1$, a contradiction. Thus, we have that each latin square in $S - \{A\}$ contributes a distinct element to D . This means that there are exactly $p_1 - 1$ elements in $S - \{A\}$ and that $D = \{1, 2, \dots, p_1 - 1\}$. Therefore, $\beta_1 \alpha_1^{-1} \text{ mod } p_1 \in D$. So for some latin square $K = (\| l_{ij} \|, \alpha_k, \beta_k)$ we have that $\beta_1 \alpha_1^{-1} - \beta_k \alpha_k^{-1} \equiv \beta_1 \alpha_1^{-1} \pmod{p_1}$. However, this implies that $\beta_k \alpha_k^{-1} \equiv 0 \pmod{p_1}$, which is a contradiction because by lemma 2.2, K is not a latin square. \square

Theorem 2.11. *Suppose that the prime factorization of n is $n = p_1 p_2 \dots p_h$ such that*

$p_1 \leq p_2 \leq \dots \leq p_h$ and p_1, p_2, \dots, p_h are prime. Then there exists a maximal set of $p_1 - 1$ mutually orthogonal latin squares of order n that generate a linear code modulo n .

Proof. Let α be an integer in the range $0 < \alpha < n$ that is relatively prime to n . Then the $p_1 - 1$ latin squares of the form $L_k = (\| l_{ij}^k \|, \alpha, k)$ as k ranges from 1 to $p_1 - 1$ are, Corollary 2.9(3), mutually orthogonal. By Theorem 2.10, this is a maximal set of mutually orthogonal latin squares of order n that generate a linear code modulo n . \square

When n is a prime number, it is well known that there are exactly $n - 1$ mutually orthogonal latin squares of order n . It is worthwhile to note that in such a case, we know, by Theorem 2.11, that there exist $n - 1$ such mutually orthogonal latin squares of order n that also generate a linear code modulo n .

3.3. Examples

Example 2.12. We give an example of a linear code generated from 4 mutually orthogonal latin squares of order 5. We use the method described in the proof of Theorem 2.11 with $\alpha = 4$:

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

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

The code C generated by these latin squares is $C =$

$$\begin{aligned} & (0, 0, 0, 0, 0, 0), (0, 1, 4, 4, 4, 4), (0, 2, 3, 3, 3, 3), (0, 3, 2, 2, 2, 2), (0, 4, 1, 1, 1, 1), \\ & (1, 0, 1, 2, 3, 4), (1, 1, 0, 1, 2, 3), (1, 2, 4, 0, 1, 2), (1, 3, 3, 4, 0, 1), (1, 4, 2, 3, 4, 0), \\ & (2, 0, 2, 4, 1, 3), (2, 1, 1, 3, 0, 2), (2, 2, 0, 2, 4, 1), (2, 3, 4, 1, 3, 0), (2, 4, 3, 0, 2, 4), \\ & (3, 0, 3, 1, 4, 2), (3, 1, 2, 0, 3, 1), (3, 2, 1, 4, 2, 0), (3, 3, 0, 3, 1, 4), (3, 4, 4, 2, 0, 3), \\ & (4, 0, 4, 3, 2, 1), (4, 1, 3, 2, 1, 0), (4, 2, 2, 1, 0, 4), (4, 3, 1, 0, 4, 3), (4, 4, 0, 4, 3, 2) \end{aligned}$$

This code is linear and one example of this is as follows:

$$(1, 2, 4, 0, 1, 2) + (3, 4, 4, 2, 0, 3) + (2, 1, 1, 3, 0, 2) \\ + (3, 3, 0, 3, 1, 4) = (4, 0, 4, 3, 2, 1) \in C$$

Example 2.13. We can easily develop a formula for computing pairs of orthogonal latin squares that generate a linear code modulo n , for any odd n using Corollary 2.9:

$$L_1 = \| l_{ij} \| \text{ defined by } l_{ij} = (2^k i + j) \text{ mod } n$$

and

$$L_2 = \| l_{ij} \| \text{ defined by } l_{ij} = (2^{k-1} i + j) \text{ mod } n$$

This works whenever $2^k < n$ because $L_1 = (\| l_{ij} \|, 1, 2^k)$ and $L_2 = (\| l_{ij} \|, 1, 2^{k-1})$. However, by Corollary 2.9(3), these are orthogonal because $\gcd(2^k - 2^{k-1}, n) = \gcd(2^{k-1}, n) = 1$, since n is odd. Note that this degenerates to a well-known method of computing pairs of mutually orthogonal latin squares when k is set to 1.

References

- [1] J. Denes and A.D. Keedwell, *Latin Squares and Their Applications*, Academic Press, New York and London, (1974).
- [2] Elizabeth J. Billington and Peter J. Robinson, A list of balanced ternary designs with $R \leq 15$, and some necessary existence conditions, *Ars Combinatoria*, **16** (1983), 235-258.
- [3] T. Evans, Embedding incomplete latin squares, *American Mathematical Monthly* **67** (1960), 958-961.
- [4] John B. Fraleigh, *A First Course in Abstract Algebra*, Addison-Wesley, Massachusetts, (1997).
- [5] D. Fujitake, S. Kageyama, and T. Shimata, A class of error-correcting codes through BIB designs, *Bulletin of the ICA*, **19** (1997), 121-124.
- [6] S. Kadowaki, S. Kageyama, M. Kimura, and Y. Yanagida, Error-correcting non-binary codes through Latin squares, *Bulletin of the ICA*, **29** (2000), 67-70.
- [7] Richard E. Klima, Neil Sigmon, and Ernest Stitzinger, *Applications of Abstract Algebra with Maple* CRC Press, New York, (2000).