

**Error-correcting codes from 2-representations of the  
unitary group  $U(3,3)$**

by

TAPIWANASHE GIFT NYIKADZINO

Dissertation

Submitted in fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

MATHEMATICS

in the

FACULTY OF SCIENCE AND AGRICULTURE

School of Mathematics and Computer Sciences

at the

UNIVERSITY OF LIMPOPO

Supervisor: Dr. A. Saeidi

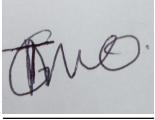
Co-Supervisor: Prof. T.T. Seretlo

2024.

## Declaration

I declare that the dissertation hereby submitted to the University of Limpopo, for the degree of Master of Science in mathematics has not previously been submitted by me for a degree at this or any other university; that it is my work in design and in execution, and that all material contained herein has duly acknowledged.

Signed: \_\_\_\_\_

A handwritten signature in black ink, appearing to be 'T.M.O.', written over a grey rectangular background.

Date: August 21, 2023

## Acknowledgments

This good work was accomplished with the contribution of many people in different ways. To all, I appreciate their assistance and support in times of hardship and distress. I am most grateful to my supervisors Dr A. Saeidi and Prof T.T. Seretlo, for their guidance and patience for this work to be completed successfully and the effort they put for me to attend the 7<sup>th</sup> Biennial International Group Theory Conference in which I met the experts and learnt a lot of new things.

I acknowledge the support I got from my family, especially my mother Monica Mokoting and my father Thomas Mokoting for their endless support throughout my studies and life as a whole, not forgetting my special friend Tebogo Ramphele for the support she gave me throughout my postgraduate studies since 2018. My classmates David Ngoako and Jan Kekana for the academic support I got from them, I did not know that I can accomplish this much until I had you to run to whenever I got lost. My academic brother Thabo Ramalatso I appreciate you for always being by my side throughout this journey. Last but not least, special thanks to ETDP SETA for the financial support and University of Limpopo and its staff more special School of Mathematical and Computer Science staff for giving me the chance to pursue my dream with their institution.

The man above all of us, Lord We do not count you among men, you are above them all and thank you lord for the endless knowledge and wisdom. Glory, honor and dominion be unto you forever.

# Abstract

In this dissertation, we use modular representation theory to find error-correcting codes admitting finite simple group as a primitive permutation group and show that every binary linear code admitting group  $G$  as a primitive permutation group is a submodule of the permutation module of the primitive action of the group. If the Schur multiplier of the group  $G$  is trivial and  $P$  is a permutation module of degree  $n$ , then every binary linear code of length  $n$  invariant under  $G$  is a submodule of  $P$ . As an illustrative example, we select the finite simple group  $G = U(3, 3)$  which is referred to by other authors as  $PSU(3, 3)$  and identify the complete set of linear codes derived from its 2-representations. We will find the maximal subgroups of the simple group  $G = U(3, 3)$ . After finding the maximal subgroups we find the permutation representation, each permutation representation has a corresponding permutation module which we will find. Our computations are based on MAGMA.

We then classify these codes and determine their properties such as the minimum distance, minimum weight and the support and other properties. Then we will discuss whether a certain code has good error-correcting or error-detecting abilities based on their properties. In addition, we use the supports of the codes to construct certain designs that remain invariant under the action of  $U(3, 3)$  and establish connections between these designs and the corresponding linear codes.

# Symbols and abbreviations

$\mathbb{N}$	Set of natural numbers
$\mathbb{Z}$	Set of Integers
$\mathbb{R}$	Set of real numbers
$\Omega$	Finite set
$V$	Vector space
$\mathbb{F}$	Finite field
$\mathbb{F}^*$	$\mathbb{F} - \{0\}$
$\mathbb{F}_q$	Galois' field of $q$ elements
$C$	Linear code
$H \leq G$	$H$ is a subgroup of $G$
$ \Omega $	cardinality/order of a set
$G:H$	Split extension of $G$ by $H$
$G.H$	General extension
$G, H, K$	Groups
$1_G$	Identity element of group $G$
$Id$	Identity transformation.
$orb(x)$	Orbit of $x$
$G_x$	Stabilizer of $x$
$Z(G)$	Centre of a group $G$
$C_G(g)$	Centralizer of $g$
$Cl(a)$	Equivalence class

$N_G(H)$	Normalizer of $H$
$\langle X \rangle$	Group generated by set $X$
$\ker(\tau)$	Kernel of $\tau$
$\text{im}(\tau)$	Image of $\tau$
$\text{Aut}(C)$	Automorphism group of a code $C$
$[n, k, d]_q$	$q$ - ary code of length $n$ , dimension $k$ and minimum distance $d$

# Contents

- 1 Basic concepts** **4**
  - 1.1 Groups . . . . . 4
  - 1.2 Group action and permutation groups . . . . . 7
  - 1.3 Group extension and Schur multiplier . . . . . 10
  - 1.4 Vector space and modules . . . . . 11
  
- 2 Representation theory** **13**
  - 2.1 Permutation representations . . . . . 13
  - 2.2  $\mathbb{F}G$ -modules . . . . . 15
  - 2.3 Ordinary representation theory . . . . . 18
  - 2.4 Modular representation theory . . . . . 19
  
- 3 Codes and designs** **22**
  - 3.1 Codes . . . . . 22
  - 3.2 Binary linear codes . . . . . 23
  - 3.3 Decoding Schemes . . . . . 28
  - 3.4 Designs . . . . . 30
    - 3.4.1 The construction of  $t$ -designs from linear codes . . . . . 32
  
- 4 Constructions of combinatorial structures** **35**
  - 4.1  $\mathbb{F}G$ -modules and  $G$ -invariant codes . . . . . 35
    - 4.1.1 Codes from quotient modules . . . . . 35

4.1.2	Codes from maximal submodules . . . . .	36
4.1.3	Permutation codes . . . . .	37
4.2	Construction of $G$ -invariant codes. . . . .	38
<b>5</b>	<b>Codes invariant under <math>U(3,3)</math></b>	<b>42</b>
5.1	The structure of the unitary group $U(3,3)$ . . . . .	42
5.2	Permutation representations and permutation modules of $U(3,3)$ . . . . .	44
5.2.1	Representation of degree 28 . . . . .	44
5.2.2	Representation of degree 36 . . . . .	48
5.2.3	Representations of degree 63 . . . . .	57



# Introduction

Coding theory is a vital study that tries to reduce data loss caused by errors occurred in transmission caused by noise, interference, or other forces. Coding theory is a field of mathematics concerned with transmitting data across noisy channels and recovering the messages. Messages are transferred in form of binary bits [37]. We have to transmit these bits along a noisy channel in which errors occur at random, but at a predictable overall rate. The first step in the transmission across a communication channel is the process of encoding the information to be transmitted using a suitable code. The end user can receive the transmitted information after it has been decoded using the decoding capabilities of the code used. The noise in the channel can distort the message and hence the user can receive a wrong message, then the necessity of error-correcting codes begins.

Single error-correcting and double error-detecting codes were introduced by R. Hamming in 1950 [13]. R. Hamming introduced some codes, the concepts of some windows, numbers, and distance. These are known as Hamming codes, Hamming windows, Hamming numbers and Hamming distances. Hamming codes are binary linear codes that were developed by R. Hamming. They are easy to encode and decode and are  $[n, k, d]$  codes, where  $n$  is the length,  $k$  the dimension and  $d$  the minimum distance. Hamming codes are useful in detecting and correcting errors. We can also use codes to construct designs [13].

Codes acquired from 2-representations of groups of finite order have been given particular attention recently. 2-representations play a crucial role in constructing codes, especially in the context of coding theory in mathematics and computer science. 2-representations are impor-

tant in constructing codes because they provide a more advanced and flexible framework for capturing algebraic structures. This, in turn, allows for the construction of error-correcting codes with enhanced capabilities and applicability in various domains, including quantum computing, topological codes, and non-abelian group-based codes.

The link between code and designs open on to the construction of support designs. The knowledge of codes and the existence of designs in codes is useful for decoding purposes [39]. The coding theory we will discuss is of great mathematical interest and relies largely on ideas from group theory. The subject of accuracy is introduced by detection and correction of errors that occur during transmission.

In [19], [14], [33] and [26] similar research was done but in different groups, where it was shown that weight distributions of codes can be used to find designs and other structures. For each of the primitive representations. A permutation group was constructed to form the orbits of the stabilizer of a point. Further investigations of alternatives ways to constructing those codes. Some of these methods will be used in this dissertation but on different group, and see the results they yield.

It is well-known that if the code admits a transitive group as a permutation group, then the code has good error-correcting properties [26, 27]. In this case we use the unitary group which is a transitive permutation group. In this dissertation, we use a method based on the modular representation theory to construct codes from finite groups. The codes we construct are the submodules of some permutation modules of the group. The method is based on the results of [28] and enables us to prove that in some cases (for example, if the group is simple and its Schur multiplier is trivial), we can find the of all binary linear codes admitting the chosen group as a primitive permutation automorphism group. We have chosen the unitary group  $U(3, 3)$ , a simple group with trivial Schur multiplier. This group is of order 6048, having 4 maximal subgroups up to conjugation [15]. However we can generalize the results of this dissertation to any simple group with trivial Schur multiplier.

Chapter 1 and 2 of this study gives brief of basic results of group theory and representation theory, mostly studied at undergraduate level but are need for later chapters. In Chapter 3

we define and give properties of codes and designs, we talked about properties such as even and doubly even. This will help us on results when we classify the codes from our selected group. Chapter describes methods that can be used to construct those structure there are several methods that can be used and in this dissertation we chose one method to construct codes and then find designs from supports of those codes other method such as Key-Moori method 1 and Key-Moori method 2 (see [36]) can be used to construct designs and find codes from those designs.

# Chapter 1

## Basic concepts

In this section, we delve into fundamental principles of group theory which form the bedrock of our exploration into representation theory and coding theory. Group theory provides the essential language and framework for understanding the complex relationships between permutation representations and error-correcting codes. Both representation theory and coding theory rely on basic knowledge of groups, such as group actions and maximal subgroups. So it is important to know basics of group theory.

### 1.1 Groups

**Definition 1.1.1** *A group  $(G, *)$  is a set  $G$  together with the binary operation  $*$  satisfying the following conditions for  $g, h, k \in G$ .*

*i For all  $g, h \in G$ ,  $g * h \in G$ .*

*ii There exists  $1 \in G$  (called the identity or neutral element[34]) such that  $g * 1 = 1 * g = g$  for all  $g \in G$ .*

*iii For every  $g \in G$  there exist  $g^{-1} \in G$  (called the inverse of  $g$ ) such that  $g^{-1} * g = g * g^{-1} = 1$ .*

*iv For all  $g, h, k \in G$ , we have  $g * (h * k) = (g * h) * k$  (associativity).*

The order of a group  $H$  symbolized by  $|H|$  is defined to be the number of elements in  $H$ . An element  $h \in H$  is said to be of order  $n$  if  $n$  is the smallest positive integer such that  $h^n = 1$ . A finite group of order  $q$  where  $q = p^n$ , with  $p$  being prime is called a  $p$ -group.

**Definition 1.1.2** A group  $G$  is called an abelian group if  $gh = hg$  for all  $g, h \in G$ .

**Definition 1.1.3** Let  $H$  be a set. Then  $H$  is called a subgroup of  $G$  if  $H \subseteq G$  and  $H$  is itself a group with the same operation defined in  $G$ .

**Lemma 1.1.1 (Subset lemma)** A subset  $H$  of  $G$  is a subgroup of  $G$  if and only if

- i.  $H \neq \emptyset$ ,
- ii. for all  $a, b \in H$ , we have  $ab^{-1} \in H$ .

**Definition 1.1.4** The centre of a group  $G$  is a subset

$$Z(G) = \{g \in G \mid gx = xg \text{ for all } x \in G\}.$$

**Remark 1.1.2** Since  $1_G \in Z(G)$ ,  $Z(G) \neq \emptyset$ , now let  $a, b \in Z(G)$  we have  $b^{-1} \in Z(G)$  and  $ab^{-1}x = axb^{-1} = xab^{-1}$ , so we have  $ab^{-1} \in Z(G)$ . Therefore  $Z(G)$  is a subgroup of  $G$ .

A set  $X$  generates  $G$  if  $G = \langle X \rangle$ , i.e. if every element of  $G$  can be written as a finite combination from  $X$  and their inverses.

**Definition 1.1.5** A group  $G$  is cyclic if it is generated by a single element, i.e.  $G = \langle g \rangle$ , for some  $g \in G$ .

**Definition 1.1.6**  $N \leq G$  is a normal subgroup  $G$  if  $gNg^{-1} = N \forall g \in G$  and it is denoted by  $N \trianglelefteq G$ .

**Definition 1.1.7** A group  $G$  is said to be simple if it has no non-trivial normal subgroups.

**Definition 1.1.8** Let  $G$  be a group. The commutator of two elements  $a, b \in G$  is the element  $[a, b] = a^{-1}b^{-1}ab$ . The derived subgroup  $G'$  is the subgroup of  $G$  generated by all commutators i.e  $G' = \langle \{[a, b] : a, b \in G\} \rangle$  [6].

If  $G$  is abelian, then  $[a, b] = a^{-1}b^{-1}ab = a^{-1}ab^{-1}b = 1_G \cdot 1_G = 1_G$  for all  $a, b \in G$ . It follows that if  $G$  is abelian then  $G' = \{1_G\}$ .

**Definition 1.1.9** A group  $G$  is called perfect if  $G = G'$ .

**Definition 1.1.10** The centralizer of an element  $g \in G$  denoted by  $C_G(g)$  is

$$C_G(g) = \{a \in G \mid ag = ga\}.$$

The centralizer of any element  $g \in G$  defines a subgroup  $G$  [3].

**Definition 1.1.11** A normalizer of a subgroup  $H$  of  $G$  is defined as

$$N_G(H) = \{g \in G \mid gHg^{-1} = H\}.$$

This defines a subgroup of  $G$  containing  $H$ . Moreover,  $H$  is a normal subgroup of  $N_G(H)$  [3].

**Definition 1.1.12** Let  $G$  be a group, then  $a, b \in G$  are said to be conjugate in  $G$  if there exists  $g \in G$  such that  $gag^{-1} = b$ .

**Definition 1.1.13** The equivalence class that contains the element  $a \in G$  denoted by  $Cl_G(a)$  or  $C_a$  is defined by  $Cl_G(a) = \{gag^{-1} \mid g \in G\}$  and is called the conjugacy class of  $G$  containing  $a$ .

It can be easily seen that conjugacy is an equivalence relation and therefore partitions  $G$  into equivalence classes [10]. If we let  $g$  be representative of conjugacy class  $C_g$  of a finite group  $G$ , then using the following theorem we can see the relationship of conjugacy class and the centralizer of  $g$ .

**Definition 1.1.14** Let  $N$  be a normal subgroup of  $G$ , a quotient group of  $G$  is defined as  $G/N = \{Ng : g \in G\}$ .

**Theorem 1.1.3** Let  $G$  be a finite group. Then the function  $\phi : G/C_G(g) \rightarrow C_g$  given by  $\phi(xC_G(g)) = xgx^{-1}$  is bijective and so  $|C_g| = |G : C_G(g)| = \frac{|G|}{|C_G(g)|}$ .

**Proof:** See [39, Theorem 2.2.4]. □

**Definition 1.1.15** A function  $\phi : G \rightarrow H$  is a homomorphism if  $\phi(ab) = \phi(a)\phi(b)$ . The kernel of  $\phi$  denoted by  $\ker(\phi)$  is a normal subgroup of  $G$  defined by  $\ker(\phi) = \{g \in G \mid \phi(g) = 1_H\}$  where  $1_H$  is the identity in  $H$  and the image of  $\phi$  is defined by  $\text{im}(\phi) = \{\phi(g) \mid g \in G\}$  and is a subgroup of  $H$ .

**Definition 1.1.16** A homomorphism  $f$  defined in Definition 1.1.15 is called an isomorphism if  $f$  is one-to-one and onto. We say that groups  $G$  and  $H$  are isomorphic if there is an isomorphism  $f : G \rightarrow H$ . If  $G = H$  then the defined homomorphism is called an automorphism.

## 1.2 Group action and permutation groups

Let  $\Omega$  be a set, a permutation of  $\Omega$  is a function  $f : \Omega \rightarrow \Omega$  which is one-to-one and onto. A permutation group defined on  $\Omega$  is a set of permutations of  $\Omega$  that forms a group under composition of functions.

**Definition 1.2.1** Let  $G$  define a group and  $\Omega$  be a finite set. A group action  $G$  on  $\Omega$  is a mapping  $G \times \Omega \rightarrow \Omega$  relating  $\alpha \in \Omega$  and  $g \in G$  an element  $g\alpha \in \Omega$  with the condition that  $\forall \alpha \in \Omega$  and  $\forall g, h \in G$ ,  $\alpha^1 = \alpha$ ,  $(hg)\alpha = h(g\alpha)$ .

**Definition 1.2.2** Given an action of  $G$  on a set  $\Omega$ , the orbit of  $\alpha \in \Omega$  denoted by  $\text{Orb}(\alpha) = \{g\alpha \mid g \in G\}$  and is a subset of  $\Omega$ . A stabilizer  $G_\alpha$  of  $\alpha$  is a set  $G_\alpha = \{g \in G \mid g\alpha = \alpha\}$ , i.e. the set of elements of  $G$  which leave  $\alpha$  fixed and is a subset of  $G$ .

**Example 1.2.1** For any two elements  $g, x \in G$  let  $g.x = gxg^{-1}$ . This defines a group action of  $G$  on  $G$ . The orbits of this action are conjugacy classes of  $G$ , and the stabilizer of a given element is the centralizer of the element. Similarly, we can define a group action of  $G$  on the set of all subsets of  $G$ , by  $g.S = gSg^{-1}$  or on the set of the conjugacy subgroups of  $G$ .

**Definition 1.2.3** *The symmetric group on a set  $\Omega$  is the group  $S_\Omega$  of all permutation of set  $\Omega$ . A group  $A_\Omega$  is an alternating subgroup of  $S_\Omega$  containing all the even permutations in  $S_\Omega$ .*

A permutation group  $G$  on a set  $\Omega$  with  $|\Omega| = n$  is a subgroup of symmetric group  $S_n$ , and  $G$  is said to be *transitive* on  $\Omega$  if, for all  $\alpha, \beta \in \Omega$ , there exists an element  $g \in G$  such that the image  $g\alpha$  of  $\alpha$  under  $g$  is equal to  $\beta$  [33]. In a natural way, an action defines a permutation representation of  $G$  on set  $\Omega$  which is homomorphism from  $G$  into  $S_n$ . Conversely a permutation representation naturally defines an action of  $G$  on  $\Omega$ .

**Definition 1.2.4** *A group action on a set  $\Omega$  is transitive if the set is non-empty and there is exactly one orbit.*

*Similarly a group action is said to be transitive on  $\Omega$  if for all  $\alpha, \beta \in \Omega$ , there exist  $g \in G$  such that  $g\alpha = \beta$ .*

**Definition 1.2.5** *Suppose  $G$  acts on a finite set  $\Omega$ . Let  $|\Omega| = n$  and  $k$  be a positive integer.  $G$  is said to be  $k$ -transitive on  $\Omega$  if every two ordered  $k$ -tuples,  $(\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_k)$  and  $(\beta_1, \beta_2, \beta_3, \dots, \beta_k)$  with  $\alpha_i \neq \alpha_j$  for all  $i \neq j$  there exist  $g \in G$  such that  $\alpha_i^g = \beta_i$  for  $i = 1, 2, 3, \dots, k$ .*

The symmetric group  $S_n$  acts  $n$ -transitively on  $\Omega = \{1, 2, 3, \dots, n\}$ .

**Definition 1.2.6** *The general linear group of degree  $n$  over a field  $\mathbb{F}$  denoted by  $GL(n, \mathbb{F})$  is the set of  $n \times n$  invertible matrices with entries from  $\mathbb{F}$ , with matrix multiplication as the group operation.*

**Definition 1.2.7** *Suppose  $V$  is a finite vector space over a finite field  $\mathbb{F}$ . The set of all invertible transformations from  $V$  to  $V$  is denoted by  $GL(V)$  and has the group structure under composition of transformation with the identity element being identity transformation  $Id(x) = x$  for all  $x \in X$ .*

More generally both  $GL(V)$  and  $GL(n, \mathbb{F})$  are both abstract automorphism groups, where in  $GL(V)$  the elements are not necessarily written as matrices, but we have that, if  $V$  is a vector over  $\mathbb{F}$  then  $GL(V) \cong GL(n, \mathbb{F})$ .



**Theorem 1.2.1** *A transitive group action of  $G$  on a subgroup  $H$  is equivalent to the action of  $G$  on a set of cosets  $G/H$  and is a quotient group if  $H \triangleleft G$ .*

**Proof:** See [14, Theorem 2.6]. □

**Definition 1.2.8** *The automorphism group  $\text{Aut}(G)$  of a group  $G$ , is the group of all automorphisms of  $G$ .*

**Definition 1.2.9** *A permutation group  $G$  is said to be primitive on  $\Omega$  if  $G$  is transitive on  $\Omega$  and the only  $G$ -invariant partitions of  $\Omega$  are the trivial partitions. Also  $G$  is imprimitive on  $\Omega$  if  $G$  preserves some non-trivial partition on  $\Omega$ .*

**Theorem 1.2.2** *(Characterization of primitive permutation groups) Suppose  $G$  is a transitive permutation group on set  $\Omega$ , then  $G$  is primitive if and only if for all  $\omega \in \Omega$  the stabilizer  $G_\omega$  is a maximal subgroup.*

**Proof:** See [7, Theorem 1.6.5]. □

**Theorem 1.2.3 (Classification of finite simple groups)** *Every finite simple group is isomorphic to one of the following groups:*

- i A group of prime order.*
- ii An alternating group of degree  $n \geq 5$ .*
- iii One of the groups of Lie type.*
- iv One of the 26 sporadic groups.*

**Proof:** See [40, Theorem 4.6]. □

The groups of prime order are the abelian simple groups. Finite groups of Lie type are the finite analogues of the semisimple lie groups. Each group in the first three classes of groups is a member of one or more infinite families of finite simple groups. There are however 26 finite simple groups which are not members of any finite family and they are called sporadic groups.

### 1.3 Group extension and Schur multiplier

A sequence  $\dots \xrightarrow{\alpha_{n-2}} G_{n-1} \xrightarrow{\alpha_{n-1}} G_n \xrightarrow{\alpha_n} G_{n+1} \xrightarrow{\alpha_{n+1}} \dots$  of groups  $G_i$  with homomorphisms  $\alpha_i$  is said to be an exact sequence at  $G_n$  if  $\text{image}(\alpha_{n-1}) = \text{ker}(\alpha_n)$ . The sequence is said to be exact if it is exact at each  $G_n$ .

**Definition 1.3.1** A finite sequence of the type  $I \rightarrow H \xrightarrow{\alpha} G \xrightarrow{\beta} K \rightarrow I$  with  $I$  meaning the trivial group is called the short exact sequence. Thus to say that the above sequence is exact is to say  $\alpha$  is injective,  $\beta$  is onto and  $\text{im}(\alpha) = \text{ker}(\beta)$ . In particular  $\alpha(H)$  is a normal subgroup of  $G$  such that  $\beta$  induce an isomorphism from  $G/\alpha(H)$  to  $K$  [29].

**Definition 1.3.2** The short exact sequence in Definition 1.3.1 is called the extension of  $H$  by  $K$  or we say that  $G$  is an extension of  $H$  by  $K$ .

**Definition 1.3.3** A set  $\text{Ext}(H, k)$  is a set consisting of all isomorphism classes of groups  $E$  that are extensions of  $H$  by  $K$ .

**Definition 1.3.4** An extension  $E$  of  $H$  by  $K$  given by  $E \equiv I \rightarrow H \xrightarrow{\alpha} G \xrightarrow{\beta} K \rightarrow I$  is called a central extension if  $\alpha H \leq Z(G)$ .

In Definition 1.3.5 we define the group  $[R, F]$  by

$$[R, F] = \langle \{[r, x] = r x r^{-1} x^{-1} \mid r \in R, x \in F\} \rangle.$$

**Definition 1.3.5** Let  $1 \rightarrow R \xrightarrow{\alpha} F \xrightarrow{\beta} K \rightarrow 1$  be a free presentation of a group  $K$  then  $([R, F] \cap R)/(R, F)$  is called the Schur multiplier of  $K$  and it is denoted by  $M(K)$ . [29]

**Definition 1.3.6** A group is said to be Schur-trivial or a group with trivial Schur multiplier if it satisfies the following equivalent conditions:

- i. The Schur multiplier is the trivial group.
- ii. The homomorphism from its exterior square to its derived subgroup defined by the commutator map is an isomorphism of groups to the derived subgroup.
- iii. It is isomorphic to its own Schur covering group with the covering map being the identity map.

## 1.4 Vector space and modules

**Definition 1.4.1** A ring  $R$  is an abelian group under addition that also has another operation (multiplication) satisfying the following conditions:

- i.  $a \cdot b \in R$  for all  $a, b \in R$ .
- ii.  $a \cdot (b \cdot c) = (a \cdot b) \cdot c$  for all  $a, b, c \in R$ .
- iii.  $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$ .

**Definition 1.4.2** A field is a ring  $\mathbb{F}$  having the following conditions:

- i.  $x \cdot y = y \cdot x$  for all  $y, x \in \mathbb{F}$
- ii. There is an element  $1 \in \mathbb{F}$ ,  $1 \neq 0$  and  $x \cdot 1 = x$ .
- iii. If  $x \neq 0$  then there exist  $x^{-1}$  such that  $x \cdot x^{-1} = 1$ .

In this dissertation we use  $\mathbb{F}_q$  to denote the field with  $q$  elements  $\{0, 1, 2, \dots, q - 1\}$ .

**Definition 1.4.3** Let  $R$  be an arbitrary ring with unity, then:

1. A left  $R$ -module is an abelian group  $M$  together with scalar multiplication  $R \times M \rightarrow M$  (simply written as  $mr = rm$ ) such that for all  $r, s \in R$  and  $m, n \in M$

- i.  $r(m + n) = rm + rn$ ,
- ii.  $(r + s)m = rm + sm$ ,
- iii.  $r(sm) = (rs)m$ .

If  $R$  has a unity we also require  $1m = m$  for all  $m \in M$ .

2. A right  $R$ -module is an abelian group  $M$  together with scalar multiplication  $M \times R \rightarrow M$  such that

- i.  $(m + n)r = mr + nr$ ,

$$ii. m(r + s) = mr + ms,$$

$$iii. m(rs) = (mr)s.$$

Also if  $R$  has unity we also require  $m1 = m$  for all  $m \in M$ .

If the ring  $R$  is commutative, then left  $R$ -modules and right  $R$ -module have the same structure and we call it  $R$ -modules (see[18]).

**Definition 1.4.4** *If we let  $R$  in Definition 1.4.3 to be a field  $\mathbb{F}$ , then an  $\mathbb{F}$ -module  $V$  is called a vector space over  $\mathbb{F}$ .*

The elements of a vector space  $V$  are called vectors. If a field  $\mathbb{F}$  is of order  $q$  we denote a vector with elements of order  $n$  space over  $\mathbb{F}_q$  by  $\mathbb{F}_q^n$ .

A subset  $S$  of vector space  $V$  over a field  $\mathbb{F}$  is said to be linearly independent if given any subset  $\{x_1, x_2, x_3, \dots, x_n\}$  of  $S$ ,  $x_i \neq x_j$  for  $i \neq j$

$$a_1x_1 + a_2x_2 + a_3x_3 + \dots + a_nx_n = 0$$

implies  $a_i = 0$  for all  $i$ .

**Definition 1.4.5** *A subset  $S$  of a vector space  $V$  is called a basis of  $V$  if it contains linearly independent vectors and it generates  $V$  i.e any vector in  $V$  can be written as a linear combination of finite elements of  $S$ .  $S$  is said to span  $V$  and is denoted  $\text{Span}(V)$ .*

**Definition 1.4.6** *If  $V$  is spanned by a finite set, then  $V$  is said to be finite dimensional, and the dimension of  $V$ , written as  $\dim(V)$ , is the number of vectors in a basis for  $V$ . The dimension of the zero vectors space  $\{0\}$  is defined to be zero. If  $V$  is not spanned by any finite set, then  $V$  is said to be infinite dimensional.*

# Chapter 2

## Representation theory

Representation theory is a fundamental branch of mathematics that focuses on the matrix representation of groups. It allows us to represent abstract mathematical groups by invertible matrices. By representing groups a matrix we gain insight onto structure, symmetry and actions of the group. In this chapter, we brief on concepts of group representations and explore relationship between groups and the general linear group,  $GL(n, \mathbb{F})$ . Throughout this chapter and all subsequent sections, it is important to note that we work with  $\mathbb{F}$  as a finite field, where our vector space  $V$  is finite over  $\mathbb{F}$ .

### 2.1 Permutation representations

**Definition 2.1.1** *A matrix representation of a group  $G$  over a field  $\mathbb{F}$  is a homomorphism  $\rho : G \rightarrow GL(n, \mathbb{F})$ .*

**Definition 2.1.2** *A permutation representation of a group  $G$  is a homomorphism  $\phi : G \rightarrow S_G$ .*

If  $G$  act on  $X$ , and  $|X| = n$ , then  $G \rightarrow S_X \cong S_n \rightarrow GL(V)$ . This leads directly to the permutation representation of  $G$ . Theorem 2.1.1 gives us a method of constructing such representations.

**Theorem 2.1.1 (Cayley)** *Every group  $G$  is isomorphic to a subgroup of  $S_G$ . In particular if  $|G| = n$ , then  $G$  is isomorphic to a subgroup of a symmetric group  $S_n$ .*

**Proof:** For each  $x \in G$ , define  $T_x: G \rightarrow G$  by  $T_x(g) = xg$ . Then  $T_x$  is one to one and onto; so that  $T_x \in S_G$ . Now if we define  $\tau: G \rightarrow S_G$  by  $\tau(x) = T_x$ , then  $\tau$  is a monomorphism. Hence  $G \cong \text{Image}(\tau) \leq S_G$ .  $\square$

**Remark 2.1.2** *The homomorphism  $\tau$  defined in Definition 2.1.1 is called the left regular representation of  $G$ .*

**Theorem 2.1.3** *Let  $GL(n, \mathbb{F})$  denote the general linear group over a field  $\mathbb{F}$ . If  $G$  is a finite group of order  $n$ , then  $G$  is isomorphic to a subgroup of  $GL(n, \mathbb{F})$ .*

**Proof:** Let  $T_x$  be as in Cayley's theorem (see Theorem 2.1.1). Assume that  $G = \{g_1, g_2, g_3, \dots, g_i\}$ . Let  $P_x = (a_{ij})$  denote the  $n \times n$  matrix given by

$$a_{ij} = \begin{cases} 1_{\mathbb{F}}, & \text{if } T_x(g_i) = g_j \\ 0_{\mathbb{F}}, & \text{otherwise.} \end{cases}$$

Then  $P_x$  is a permutation matrix, that is a matrix obtained from the identity matrix by permuting its columns. Define  $\pi: G \rightarrow GL(n, \mathbb{F})$  by  $\pi(x) = P_x$ . Then  $\pi$  is a monomorphism.  $\square$

A permutation matrix is a matrix in which every row and every column has a unique non-zero entry and all non-zero entries are equal to 1 [35].

**Example 2.1.1** *Let  $\tau: G \rightarrow GL(m, \mathbb{F})$  and  $\psi: G \rightarrow GL(n, \mathbb{F})$  be representations of  $G$  over  $\mathbb{F}$ . We say that  $\tau$  is equivalent to  $\psi$  if  $n = m$  and there exists an invertible  $n \times n$  matrix  $T$  such that for all  $g \in G$ ,  $\tau g = T^{-1}(\psi g)T$ .*

**Definition 2.1.3** *A representation of  $G$  (over  $\mathbb{F}$ ) is a homomorphism  $G \rightarrow GL(V)$ . Where  $V$  is a vector space over  $\mathbb{F}$ .*

Suppose  $\dim(V)$  is finite. Let  $(v_1, v_2, v_3, \dots, v_n)$  be an ordered basis for  $V \cong \mathbb{F}^n$ , which means  $GL(V) \cong GL(n, \mathbb{F})$ . This means  $\rho: G \rightarrow GL(V) \cong GL(n, \mathbb{F})$  gives a matrix representation.

Conversely,  $GL(n, \mathbb{F}) \cong GL(\mathbb{F}^n)$ , so matrix representations give representation on  $\mathbb{F}^n$ . The other way to define a representation in terms of  $\mathbb{F}$ -linear invertible maps, is an action on by linear maps  $gv = \rho(g)(v)$  for  $g \in G$ .

**Definition 2.1.4** Suppose  $\phi : G \rightarrow GL(n, \mathbb{F})$  is a representation of  $G$  on a vector space  $V = \mathbb{F}^n$  and suppose  $W$  is a subspace of  $V$  of dimension  $m$  such that  $\phi_g(W) \subseteq W$  for all  $g \in G$ , then the map  $\phi : G \rightarrow GL(m, \mathbb{F})$  is a representation of  $G$  known as the subrepresentation.

The subspace  $W$  in the above definition is said to be  $G$ -invariant.

**Definition 2.1.5** Let  $\psi : G \rightarrow GL(n, \mathbb{F})$  be a representation of  $G$  over  $\mathbb{F}$ . The function  $\chi$  defined by  $\chi(g) = \text{tr}(\psi(g))$  is called a character of  $\psi$ .

## 2.2 $\mathbb{F}G$ -modules

**Definition 2.2.1** Let  $V$  be a vector over  $\mathbb{F}$  and let  $G$  be a group. Then  $V$  is an  $\mathbb{F}G$ -module if a multiplication  $vg$  ( $v \in V$  and  $g \in G$ ) is defined, satisfying the following conditions  $\forall u, v \in V, g, h \in G$  and  $\lambda \in \mathbb{F}$ :

i.  $vg \in V$ .

ii.  $v(gh) = (vg)h$ .

iii.  $v1_G = v$ .

iv.  $(\lambda v)g = \lambda(vg)$ .

v.  $(u + v)g = ug + vg$ .

Let  $V$  be an  $\mathbb{F}G$ -module and let  $\mathcal{B}$  be a basis of  $V$ , for each  $g \in G$  let  $[g]_{\mathcal{B}}$  denote the matrix of the endomorphism  $v \rightarrow vg$  of  $v$  relative to basis  $\mathcal{B}$ .

The connection between  $\mathbb{F}G$ -modules and representations of  $G$  over  $\mathbb{F}$  is given in the following basic result.

**Theorem 2.2.1** (1) If  $\psi : G \rightarrow GL(n, \mathbb{F})$  is a representation of  $G$  over  $\mathbb{F}$ , and  $V = \mathbb{F}^n$ , then  $V$  becomes an  $\mathbb{F}G$ -module if we define the multiplication  $vg$  by

$$vg = v(g\psi) \quad (v \in V, g \in G).$$

Moreover, there is a basis  $\mathcal{B}$  of  $V$  such that

$$g\psi = [g]_{\mathcal{B}} \quad (g \in G)$$

(2) Assume that  $V$  is an  $\mathbb{F}G$ -module and let  $\mathcal{B}$  be a basis of  $V$  then function  $g \rightarrow [g]_{\mathcal{B}}$  ( $g \in G$ ) is a representation of  $G$  over  $\mathbb{F}$ .

**Proof:** See [25, Theorem 4.12]. □

**Example 2.2.1** Let  $V = \mathbb{R}^4$

$$e_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, e_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, e_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$P_a \cdot e_1 = e_2, P_a \cdot e_2 = e_1, P_a \cdot e_3 = e_4, P_a \cdot e_4 = e_3.$

**Definition 2.2.2** Let  $G$  be a subgroup of  $S_n$ . The  $\mathbb{F}$ -module  $V$  with basis  $v_1, v_2, \dots, v_n$  such that  $v_i g = v_{ig}$  for all  $i \in \{1, 2, 3, \dots, n\}$  and all  $g \in G$  is called permutation modules for  $G$  over  $\mathbb{F}$ . We call  $v_1, v_2, \dots, v_n$  natural basis of  $V$ . In particular if  $\mathbb{F} = \mathbb{F}_q$  is a finite field of order  $q$ , where  $q$  is a power of prime  $p$  and  $G$  be a finite group acting primitively on a finite set  $\Omega$ . Define  $V = \mathbb{F}\Omega$  to be the vector space of  $\mathbb{F}$  of all linear combinations of  $\sum \lambda_i x_i, \lambda_i \in \mathbb{F}$  and  $x \in \Omega$  i.e. the vector space with basis of set  $\Omega$ . To define an  $\mathbb{F}G$ -module on  $V$  it suffices to stipulate the action of the elements of  $G$  on the basis element of  $V$ . So we can consider the group  $\rho : G \rightarrow GL(V)$  defined by  $\rho(g, x) \mapsto x^g = \rho(g)(x) \quad g \in G, x \in V$ . Extending



linearly the induced  $G$ -action on  $V$  makes  $V$  into an  $\mathbb{F}G$ -module known as  $\mathbb{F}\Omega$ -permutation module.

**Example 2.2.2** Let  $g = (12)(34)$  and  $v = a_1e_1 + a_2e_2 + a_3e_3 + a_4e_4 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{pmatrix}$

$$g.v = \begin{pmatrix} a_2 \\ a_1 \\ a_4 \\ a_3 \end{pmatrix}$$

We notice that the results can be obtained by multiplying  $v$  by the matrix  $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ .

Also

$$B = \left\{ g.e_1 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}, g.e_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, g.e_3 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, g.e_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \right\}.$$

**Theorem 2.2.2** Let  $\mathbb{F}$  be a field and  $G$  be a finite group, then there is a bijective correspondence between finitely generated  $\mathbb{F}G$ -modules and representations of  $G$  on a finite dimensional  $\mathbb{F}$ -vector spaces (see [25, Summary of chapter 4]).

**Proof:** Suppose  $\rho : G \rightarrow GL(V)$  is a homomorphism, then it is clear that the action of  $G$  on  $V$  defined by setting  $gv = \rho(g)(v)$  is linear. Conversely, if we have a linear action of  $G$  on  $V$ , then without loss of generality we can define a homomorphism  $\rho : G \rightarrow GL(V)$  by  $\rho(g)(v) = gv$ . These processes are evidently mutually inverse, establishing the desired correspondence.  $\square$

**Definition 2.2.3** Let  $V$  be an  $\mathbb{F}G$ -module, a subspace  $U$  of  $V$  which is also an  $\mathbb{F}G$ -module is called an  $\mathbb{F}G$ -submodule of  $V$ .

**Definition 2.2.4** A non-empty  $\mathbb{F}G$ -module  $V$  is said to be simple or irreducible if it has only the trivial submodules, and it is called reducible if otherwise.

A reducible  $G$  module is either completely reducible or indecomposable based on whether  $V$  can or cannot be expressed as direct sum of its simple submodules.

**Definition 2.2.5** A module  $M$  is said to be completely reducible or semisimple if it can be written as a direct sum of irreducible submodules.

**Remark 2.2.3** Every permutation module has a corresponding representation. We notice that there is a one-to-one correspondence between representations of  $G$  and  $\mathbb{F}G$ -modules. A permutation representation is irreducible if and only if it corresponds with an irreducible  $\mathbb{F}G$ -module [25].

**Proof:** Given a representation  $\rho : G \rightarrow GL(V)$ , we define module action  $\mathbb{F}G$  on  $V$  by  $\sum gv = \sum \rho(g)v$ .

Conversely: Given an  $\mathbb{F}G$ -module  $V$ , the linear map  $\rho(g) : v \rightarrow gv$  is an automorphism of  $V$  and  $\rho(g_1)\rho(g_2) = \rho(g_1g_2)$  so  $\rho : G \rightarrow GL(V)$  is a representation.

## 2.3 Ordinary representation theory

In ordinary representation theory the classification of representations is based on description of irreducible  $\mathbb{F}G$ -modules of a given groups. Maschke's theorem gives the conditions under which  $\mathbb{F}G$ -module is semisimple.

**Theorem 2.3.1 (Maschke's theorem)** Let  $G$  be a finite group and  $\mathbb{F}$  a field whose characteristics is known to be 0 or a prime  $p$  such that  $p \nmid |G|$ . Then every  $\mathbb{F}G$ -module  $V$  is completely reducible i.e. if  $V$  is an  $\mathbb{F}G$ -module and  $U$  any submodule of  $V$ , then there exists a submodule  $W$  of  $V$  such that  $V = U \oplus W$ . In particular, the group algebra  $\mathbb{F}G$  is semisimple. □

**Proof:** See [41, Theorem 1.2.1].

## 2.4 Modular representation theory

Unfortunately Maschke's theorem fails if the characteristic divides  $|G|$ . Therefore in modular representation theory we may face indecomposable modules which are not irreducible. The problem arises because not all finite dimensional  $\mathbb{F}G$ -module are completely irreducible. Let  $G$  be a finite group of order  $n$ ,  $\mathbb{F}$  a field with characteristics  $p > 0$  where  $p$  is prime and  $V$  be an  $\mathbb{F}$  vector space. Then we define a linear  $G$  representation of  $V$  over a field  $\mathbb{F}$  as a homomorphism  $\rho : G \rightarrow GL(V)$ [42]. The representation  $\rho$  is called modular if  $p \mid |G|$ . The vector space  $V$  becomes a  $G$ -module if we define  $gv = \rho(g)(v)$  for  $g \in G$  and  $v \in V$ .

**Definition 2.4.1** *Let  $V \neq \emptyset$  be a permutation module.  $V$  is said to be simple or irreducible if it has only trivial submodules and is reducible otherwise. It is decomposable if it can be expressed as the direct sum of two non-trivial submodules and indecomposable otherwise.*

The difference distinctness ordinary and modular representation theory is highlighted in Theorem 2.4.1.

**Theorem 2.4.1** *Every finitely generated  $\mathbb{F}G$ -module over group algebra  $\mathbb{F}G$  is semisimple if and only if the  $\text{char}(\mathbb{F}) \nmid |G|$ .*

**Proof:** See [3, Proposition 3.1]. □

It follows that if the characteristic  $p$  of the field  $\mathbb{F}$  divides  $|G|$ , then Maschke's theorem does not apply.

**Theorem 2.4.2 (Krull-Schmidt Theorem)** *If the module  $M$  can be written as*

$$M = W_1 \oplus W_2 \oplus W_3 \oplus \dots \oplus W_l$$

*and  $M = U_1 \oplus U_2 \oplus U_3 \oplus \dots \oplus U_n$  where  $W_i$  and  $U_j$  are indecomposable, then  $l = n$  and  $W_i \cong U_j$ .*

**Proof:** See [32, Theorem 5]. □

**Theorem 2.4.3** *If  $G$  is a  $p$ -group and  $\mathbb{F}$  is a field of characteristic  $p$ . Then  $\mathbb{F}G$  is indecomposable.*

**Proof:** See [38, Lemma 3.3]. □

**Theorem 2.4.4** *If  $G$  is a finite group and  $\mathbb{F}$  is a field whose characteristic does not divide  $|G|$ , then every finitely generated  $\mathbb{F}G$ -module is completely reducible (equivalently, every  $\mathbb{F}$ -representation of  $G$  of finite degree completely reducible).*

**Proof:** See [21, (Chapter 18.1), Corollary 2]. □

**Theorem 2.4.5** *Let  $G$  be a group and  $|G| = q \cdot p^\alpha$  such that  $\gcd(p, q) = 1$ . Let  $\mathbb{F}$  be a field of characteristic  $p$ . Then  $G$  possesses finitely generated  $\mathbb{F}G$ -module which are completely reducible.*

**Proof:** See [38, Theorem 3.4]. □

As Theorem 2.4.5 should make it clear, it is very much not a reasonable strategy to attempt to get information about a modular representation just by trying to decompose it into simple modules. This is a useful technique for determining the behavior of a semisimple representation, as they decompose into a unique sum of simple modules.

**Definition 2.4.2** *Let  $G$  be a group and  $p$  any prime. Any element of  $g \in G$  can be written as  $g = st$  so that  $p$  does not divide the order of  $s$  and the order of  $t$  is a power of  $p$ . Then  $s$  is called  $p$ -regular and  $t$  is  $t$ -singular.*

**Theorem 2.4.6 (Brauer-Nesbitt)** *Let  $G$  be a group of order  $g = p^{a\alpha}$ ,  $p$  a prime and  $(q, p) = 1$ . An irreducible representation  $Z_i \equiv 0 \pmod{p^a}$  remains irreducible as a modular representation .*

**Proof:** See [12, Theorem 1]. □

**Definition 2.4.3** *A composition series for an  $\mathbb{F}G$ -module  $V$  is a series of submodules of the form  $V = V_0 \supseteq V_1 \supseteq V_2 \supseteq \cdots \supseteq V_t = 0$  such that each  $i \geq 1$  the factor  $V_{i-1}/V_i$  is irreducible. The integer  $t$  is called the length of the module  $V$ . If  $t$  is infinite then we say  $V$  has no composition series.*

**Theorem 2.4.7 (Jordan-Holder Theorem for  $\mathbb{F}G$ -modules)** *If  $V$  is a finite dimensional  $\mathbb{F}G$ -module then,  $V$  possess a composition series and the composition factors are independent of the choice of factor series.*

**Proof:** See [12, Corollary 8.7]. □

The consequence of the Jordan-Holder Theorem is that any two composition series factors  $M_i/M_{i+1}$  of one of the series are simply a permutation of the composition factors of the other.

**Proposition 2.4.8** *Let  $X$  be  $G$ -set and  $P$  be a primitive permutation module over a finite field  $\mathbb{F}$  with respect to the action of  $G$  on  $X$ . Then  $P$  contains submodules  $S_1$  and  $S_2$  of degree 1 and  $|X| - 1$ , respectively.*

**Proof:** See [19]. □

# Chapter 3

## Codes and designs

In this chapter, our primary focus is on the concepts of codes and designs, along with a careful exploration of the inherent properties that define these mathematical structures. Also  $q$  denotes a power of prime  $p$  and  $\mathbb{F}_q$  denotes a field with  $q$  elements. For a positive integer  $n$ ,  $\mathbb{F}_q^n$  is a vector space of dimension  $n$ , consisting of vectors made of elements in  $\mathbb{F}_q$ .

### 3.1 Codes

**Definition 3.1.1** *Let  $\mathbb{F}_q$  be a finite field or an alphabet of  $q$ -elements. A  $q$ -ary code  $C$  is a set of finite sequence of symbols of  $\mathbb{F}_q$ , called a codeword, and written  $x_1x_2x_3\dots x_n$  or  $(x_1, x_2, x_3, \dots, x_n)$ , where  $x_i \in \mathbb{F}_q$  for  $i = 1, 2, 3, \dots, n$ . If all sequences have the same length  $n$ , then  $C$  is called a block code of block length  $n$ . The number of elements in  $C$ , denoted by  $|C|$ , is called the size of the code. A code of length  $n$  and size  $M$  is called an  $(n, M)$ -code. A code over  $A = \{0, 1\}$  is called a binary code and a code over  $B = \{0, 1, 2\}$  is called a ternary code[31].*

In this dissertation we will mainly focus on binary linear codes.

## 3.2 Binary linear codes

**Definition 3.2.1** A linear code with length  $n$  over  $\mathbb{F}_q$  is a vector subspace of  $\mathbb{F}_q^n$ .

**Definition 3.2.2** A linear binary code  $C$  is a subspace of  $\mathbb{F}_2^n$ . We call it  $[n, k]$ -code where  $n$  is the length of a code and  $k$  is its dimension. Codewords are vectors of  $C$ .

**Theorem 3.2.1** Suppose  $G$  is a finite group and  $\Omega$  a finite  $G$  set. Then the  $\mathbb{F}_2G$  submodules of  $\mathbb{F}\Omega$  are precisely the  $G$ -invariant codes (i.e.  $G$ -invariant subspaces of  $\mathbb{F}\Omega$ ) [14].

**Proof:** Suppose  $G$  is a finite permutation group acting on a set  $\Omega$ . Let  $V = \mathbb{F}\Omega$  be the  $\mathbb{F}$  vector space with basis the elements of  $\Omega$ . Let  $\rho : G \rightarrow GL(V)$  be a representation of  $G$  given by  $\rho(g)(x) = g(x) \forall g \in G$  and  $x \in \Omega$ . We can consider  $V$  as the  $\mathbb{F}_2G$ -module obtained from  $\rho$ . Let  $S$  be an  $\mathbb{F}_2G$ -submodule of the permutation module  $V$ . Then by Definition 3.2.15 we have  $(\sum_{g \in G} \alpha_g g) \cdot S \in \mathcal{S}$ ,  $\forall \sum_{g \in G} \alpha_g g \in \mathbb{F}_2G$  and  $S \in \mathcal{S}$ . In particular,  $g \cdot S \in \mathcal{S}$  for all  $g \in G$  and  $S \in \mathcal{S}$ . Thus, for all  $g \in G$  and  $S \in \mathcal{S}$  we obtain  $\rho(g)(S) \in \mathcal{S}$  or  $g(S) \in \mathcal{S}$  and so  $\mathcal{S}$  is  $G$ -Invariant. Conversely. If  $\mathcal{S}$  is  $G$ -invariant, then for all  $g \in G$  and  $S \in \mathcal{S}$  we have  $\rho(g)(S) \in \mathcal{S}$ . Therefore for scalars  $\alpha_g \in \mathbb{F}_2$  we have  $\sum_{g \in G} \alpha_g \rho(g)(S) \in \mathcal{S}$ . by linearity. This implies that  $(\sum_{g \in G} \alpha_g g) \cdot S \in \mathcal{S}$ . □

**Example 3.2.1** Suppose  $x \in V$  and  $X = \{i_1, i_2, i_3, \dots, i_k\} \subseteq \{1, 2, 3, \dots, n\}$  are the nonzero coordinates of  $x$ , then  $x = (1010001) = e_1 + e_3 + e_7$  is represented as  $X = \{1, 3, 7\}$ .

**Definition 3.2.3** The Hamming distance  $d(u, v)$  between vectors  $u, v \in C$  is the number of coordinates in which they differ.

**Lemma 3.2.2** The Hamming distance between any vectors is a metric on  $\mathbb{F}^n$ , i.e.

- i.  $d(v, w) = 0$  if and only if  $v = w$ ,
- ii.  $d(v, w) = d(w, v)$ , for all  $v, w \in \mathbb{F}^n$ ,
- iii.  $d(u, v) \leq d(u, v) + d(v, w)$ , for all  $u, v, w \in \mathbb{F}^n$ .

**Proof:** See [5, Proposition 2.1.1]. □

**Definition 3.2.4** Let  $V = \mathbb{F}^n$  for any vector  $v = (v_1, v_2, v_3, \dots, v_n)$ , let  $S = \{i | v_i \neq 0\}$ . Then the set  $S$  is called the support of  $v$  and the weight of  $v$  denoted by  $wt(v)$  is  $|S|$ . The minimum weight of a code  $C$ , denoted by  $wt(C)$ , is

$$wt(C) = \min\{wt(v) | v \in C, v \neq 0\}$$

*i.e. the minimum of the weights of the non-zero code-words.*

**Lemma 3.2.3** We have  $d(u, v) = wt(u - v)$  for all  $u, v \in \mathbb{F}_2^n$ .

**Proof:** Let  $u, v \in \mathbb{F}_2^n$ . By definition,  $d(u, v)$  is the number of places where  $u$  and  $v$  differ. Then, the vector  $u - v$  will have 1 precisely in the places where  $u$  and  $v$  differ and 0 in the places where they are the same. In other words,  $d(u, v)$  is equal to the number of places where there is 1 in the vector  $u - v$ . But, the number of places with 1 is, by definition, is the weight of that vector. So,  $d(u, v) = wt(u - v)$ . □

**Theorem 3.2.4** Let a linear code  $C$  be a vector space over  $\mathbb{F}_2$ . If  $\dim(C) = k$ , then  $C$  has  $2^k$  codewords.

**Proof:** Suppose  $\dim(C) = k$  and let  $\{x_1, x_2, x_3, \dots, x_k\}$  be a basis for  $C$ . Then,  $C = \lambda_1 x_1 + \lambda_2 x_2 + \lambda_3 x_3 + \dots + \lambda_k x_k \in \mathbb{F}_2$ . Since  $|\mathbb{F}_2| = 2$ , there are exactly 2 choices for each  $\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_k$ . Each choice gives a different word and so  $C$  has exactly  $2^k$  codewords. □

**Definition 3.2.5** For any code  $C$ , the minimum distance of the code, denoted  $d(C)$ , is defined by  $d(C) = \min\{d(u, v) | u, v \in C, u \neq v\}$ . An  $[n, M]$ -code of minimum distance  $d$  is called an  $[n, M, d]$ -code. The values  $n, M, d$  are called the parameters of the code, and if the field is known to be  $\mathbb{F}_q$ , then code can be presented as  $[n, M, d]_q$ .

**Remark 3.2.5** If the minimum distance of a code of length  $n$  is 0 or  $n$ , we consider the code trivial. If the minimum distance is 1 we call it a repetition code.

**Theorem 3.2.6** Let  $C$  be an  $[n, k, d]$  linear code. Then the minimum distance  $d = d(C)$  is the minimum weight of  $C$ .



**Proof:** We note that in  $\mathbb{F}^n$ ,  $d(v, w) = wt(v - w)$ . Since  $C$  is a subspace  $v - w \in C$  for any  $v, w \in C$  and the results follow.  $\square$

**Definition 3.2.6** Suppose  $C$  is a linear code, and  $A_k$  is the number of codewords of weight  $k$ . Then the weight enumerator of  $C$  is the polynomial  $\sum_{k=0}^n A_k x^{n-k} y^k$ , where  $x$  is the zero point and  $y$  is a non-zero coordinates of the code.

Clearly the coefficient of  $A_k$  is the number of vectors with weight  $k$ . Therefore the weight distribution classifies codewords according to the number of non-zero coordinates. A list of non-zero  $A_k$  is usually called the weight distribution of the code.

**Definition 3.2.7** Suppose  $C$  is a code in  $V$ , then  $C^\perp$  is the dual code or orthogonal code of  $C$  defined as  $C^\perp = \{y \in \mathbb{F}_p^n \mid x \cdot y = 0 \forall x \in C\}$ .

**Lemma 3.2.7**  $C^\perp$  is a linear code.

**Proof:** See [9].  $\square$

**Lemma 3.2.8** Let  $C$  be the repetition code of length  $n$  over finite field  $\mathbb{F}_2$  and  $C^\perp$  the dual of  $C$  then  $C = [n, 1, n]$  and  $C^\perp = [n, n - 1, 2]$ .

**Proof:** If  $w$  is a non-zero code-word in  $C$ , then  $w = (\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_n)$  where  $\lambda_i \in \mathbb{F}$ . Hence the weight of  $w$  is  $n$ . Now we note from the definition of a dual of a code we observe that the code word  $(1, |\mathbb{F}^*|, 0, \dots, 0)$  is in  $C^\perp$ . Also if  $v = (v_1, v_2, v_3, \dots, v_n)$  is a non-zero codeword of  $C^\perp$ , then  $v_1 + v_2 + v_3 + \dots + v_n = 0$ . So no codewords of weight 1 exist therefore the minimum weight of  $C^\perp$  is 2.  $\square$

**Remark 3.2.9** If  $C = [n, k, d]$  then  $C^\perp = [n, n - k, d']$  and  $d'$  is not necessarily related to  $d$ , which can be proven using Lemma 3.2.7 and the Rank-Nullity Theorem from Linear Algebra[30]. The minimum distance  $d'$  of  $C^\perp$  is called the dual distance of  $C$ .

**Definition 3.2.8** A code is called projective if any 2 of its coordinates are linearly independent i.e. it has a dual distance  $d^\perp \geq 3$

**Definition 3.2.9** *A binary code is even if the weight each of its codewords is divisible by 2.*

**Definition 3.2.10** *A binary code is doubly even if the weight of each of its codewords are divisible by 4.*

It can be seen that doubly even implies even.

**Definition 3.2.11** *A linear code is self-dual if  $C = C^\perp$  and it is self-orthogonal if  $C \subseteq C^\perp$ .*

**Lemma 3.2.10** *A binary self-orthogonal code  $C$  is even.*

**Proof:** Suppose we have the  $w = (a_1, a_2, \dots, a_n)$  where  $a_i \in \{0, 1\}$ . If it is in the self-orthogonal code, then  $w \cdot w = 0$  over the field of two elements, but  $w \cdot w = a_1^2 + a_2^2 + a_3^2 \cdots + a_n^2$  which equals the number of ones in  $w$ , that is the weight of  $w$ . So  $w \cdot w = 0$  in  $\mathbb{F}_2$  if and only if  $w$  has even weight.  $\square$

**Definition 3.2.12** *Let  $\mathbb{F}$  be a field and  $C$  be a linear code over  $\mathbb{F}$  then hull of  $C$  is the intersection of  $C$  and its dual code denoted by  $Hull(C) = C \cap C^\perp$ .*

**Lemma 3.2.11** *Let  $C$  be the code of length  $n$  over a finite field  $\mathbb{F}_q$  and  $C^\perp$  the dual of  $C$ . Then all codewords  $J$  lie in  $C$  if and only if  $q$  divides the sum of all coordinates of each  $w \in C^\perp$ . In particular if  $q = 2$ , then  $J \in C$  if and only if  $C^\perp$  is even.*

**Proof:** See [19, Lemma 2]  $\square$

It is clear that  $Hull(C)$  is also a linear code. It is easy to see that a linear code  $C$  is self-orthogonal if and only if the dimension of  $Hull(C)$  is equal to the dimension of  $C$ , i.e  $Hull(C) = C$ , and it is self-dual if and only if  $Hull(C) = C^\perp$ .

**Definition 3.2.13** *A linear code  $C$  is said to be linear complementary dual (LCD) if  $Hull(C) = \{0\}$ .*

**Corollary 3.2.12** *A code is even if and only if it is contained in the dual of the repetition code.*

**Proof:** See [19, Corollary 1]. □

**Definition 3.2.14** *Let  $C$  be a binary  $(n, k)$ -code. An automorphism of  $C$  is an element of  $S_n$  that sends codewords to codewords. The automorphism group of  $C$  is*

$$\text{Aut}(C) = \{\pi \in S_n \mid c\pi \in C \text{ for all } c \in C\}.$$

The automorphism group of  $C$  is thus a subgroup of  $S_n$  if  $C \subseteq \mathbb{F}^X$ . The existence of automorphism for  $C$  can provide a richer structure for the code and allows us to make some deeper results from algebra. This is particularly the case when  $C$  has a regular automorphism group  $G \subseteq \text{Aut}(C)$ , this means that  $G$  is transitive on  $X$  and thus  $|G| = |X| = n$  the block length of  $C$ .

**Definition 3.2.15** *If  $\mathbb{F}_q$  is a field,  $C$  a vector subspace of  $\mathbb{F}_q^n$  (linear code), and  $G$  a subgroup of linear automorphisms of  $\mathbb{F}_q^n$ , then  $C$  is said to be  $G$ -invariant if  $g(C) = C$  for all  $g \in G$ .*

**Definition 3.2.16** *Let  $C$  be an  $[n, k, d]_q$  code, we have two matrices that determine the code.*

- i. A generator matrix of  $C$  denoted by  $\mathcal{G}$  is a  $k \times n$  matrix over  $\mathbb{F}_q$  whose rows forms a basis of  $C$ .*
- ii. A parity check matrix of  $C^\perp$  denoted by  $\mathcal{H}$  is a  $(n - k) \times n$  matrix over  $\mathbb{F}_q$  whose rows forms a basis of  $C^\perp$ .*

**Theorem 3.2.13** *Let  $\mathcal{H}$  be a parity check matrix of linear code  $C$ . A linear code has minimum weight  $d$  if and only if any  $d - 1$  columns of  $\mathcal{H}$  are linearly independent and there exists some  $d$  columns that are linearly dependent.*

**Proof:** Let  $C$  be a linear code. Then  $wt(C) = d(C) = d$ . So, there must exist some codewords in  $C$  with weight  $d$ . Suppose that the vector  $u$  is a code word in  $C$  such that  $wt(u) = d$ . Since  $u \in C$  implies that  $\mathcal{H}u^T = 0$  and  $u$  has  $d$  nonzero components, then there are some  $d$  columns of  $\mathcal{H}$  that are linearly dependent. For the other side, suppose that there are  $d - 1$  linearly dependent columns in  $\mathcal{H}$ . Then there must exist a nonzero vector  $v \in C$  such that  $wt(v) = d - 1$ . This however contradicts the fact that the minimum weight of  $C$  is  $d$ . So any  $d - 1$  columns of  $\mathcal{H}$  are linearly independent. □

**Definition 3.2.17** Let  $\mathcal{H}$  be a generator matrix of  $C$ . Then the permutation automorphism group of  $C$  is the stabilizer of  $C$  in the symmetric group  $S_n$  with respect to the action on the set of the columns of  $\mathcal{H}$ . We denote the permutation automorphism group of  $C$  by  $PAut(C)$ .

**Remark 3.2.14** The permutation automorphism group of a code must be distinguished from the full automorphism group of a code, the stabilizer of the action of  $\mathbb{F}_q^* \times S_n$  sending every column of  $\mathcal{H}$  to a scalar multiple of another column. Clearly  $Aut(C) = PAut(C)$  in the binary case.

Two linear codes of the same length over field  $\mathbb{F}_q$  are said to be equivalent if each can be obtained from the other by permuting the coordinates of  $\mathbb{F}_q^n$  and multiplying each coordinate by non-zero elements of the field. They are isomorphic if each can be obtained from the other by a permutation of coordinates and this is an isomorphism between two codes. In binary linear codes the notion of equivalence and isomorphic coincide.

### 3.3 Decoding Schemes

In this section we talk about the ability of a code to detect or correct errors, how many errors can a code detect or correct. These properties can be determined using the parameters (code length, dimension and minimum distance) of a code.

**Theorem 3.3.1** Let  $C$  be a code of the minimum distance  $d$  if  $d < s + 1$  for  $s > 0$ , then  $C$  can detect up to  $s$  errors in any codeword or, if  $d \geq 2t + 1$ , then  $C$  can be used to correct up to  $t$ -errors.

**Proof:** See [5, Theorem 2.1.1]. □

**Theorem 3.3.2** Let  $C$  be a code with minimum distance  $d$ . Then  $C$  can detect up to  $d - 1$  errors and can correct  $\frac{d-1}{2}$  errors.

**Proof:** See [5, Corollary 2.1.1]. □

This means the code with larger minimum distance can detect and correct more errors.

The decoding scheme in which a received word  $y$  is decoded as the closest word in the

$q$ -ary code to  $y$ , should such a word be uniquely determined, is called nearest neighbour decoding. Here close is measured in terms of the Hamming distance between two codewords. Thus, the greater the minimum distance of a code, the larger the number of errors that can be corrected. Assuming the use of the symmetric  $q$ -ary channel, this decoding algorithm maximizes the probability that, after decoding, the correct word is finally received. Note that for large codes this algorithm is costly as it requires a comparison between the received vector  $y$  and every codeword in the code. For a linear code, the syndrome of the received vector  $y$ , denoted  $Syn(y)$ , can be used to reduce the number of comparisons that are needed and to reduce the amount of memory needed to implement nearest neighbour decoding. This method is referred to as syndrome decoding (see [14]).

From Theorem 3.3.2 we note that a code needs to have a large minimum distance to correct many errors, however code dimension needs to be small for fast transmission and code length for large number of messages but this is a conflict so it's not easy to have the code satisfying those conditions.

**Theorem 3.3.3 (Singleton Bound)** *For any code  $C$  with minimum distance  $d$ , we have  $|C| \leq q^{n-d+1}$ . Moreover for a linear code,  $[n, k, d]$ -code, this means that  $q^k \leq q^{n-d+1}$ . This turn implies that  $k \leq n - d + 1$  or  $d \leq n - k + 1$ .*

**Proof:** If we consider a code with size  $|C|$  and distance  $d$ , we know that every word differs in at least  $d$  positions. If we were to truncate the codewords by ignoring the last  $d - 1$  positions, all the new codewords must be different. So we still have  $|C|$  codewords remaining, but now we are in dimension  $n - (d - 1)$ . We know that there is total of  $q^{(n-1)+d}$  codewords of this dimension, therefore we see that  $|C| \leq q^{n-d+1}$ . This proves the result along with the knowledge that when  $C$  is linear,  $|C|$  is just the size of the  $k$ -dimensional subspace over  $\mathbb{F}_q$ , which is  $q^k$ . □

## 3.4 Designs

**Definition 3.4.1** *An incidence structure is a set  $\mathcal{D} = (\mathcal{P}, \mathcal{B}, \mathcal{I})$ , where  $\mathcal{P}$  is the point set,  $\mathcal{B}$  is the block set and  $\mathcal{I}$  is an incidence relation between  $\mathcal{P}$  and  $\mathcal{B}$ . The elements of  $\mathcal{I}$  are called flags.*

A finite incidence structure is the most basic structure of design theory. If the pair  $(\mathcal{P}, \mathcal{B})$  is in  $\mathcal{I}$  we say that  $\mathcal{P}$  is incident with  $\mathcal{B}$  or  $\mathcal{B}$  contains the point  $\mathcal{P}$ . The pair  $(\mathcal{P}, \mathcal{B})$  is called a flag if it is in  $\mathcal{I}$  and anti-flag otherwise (see [5]).

**Example 3.4.1** *Let  $\mathcal{P}$  be any set and take  $\mathcal{B}$  to any subset of the power set  $2^{\mathcal{P}}$ , the set of all subsets of  $\mathcal{P}$ . The incidence in this case is defined by  $(\rho, \mathcal{B}) \in \mathcal{I}$  if and only if  $\rho \in \mathcal{B}$ .*

We will be concerned almost exclusively with those structures that have a particular degree of regularity and that are traditionally called block designs. We will simply call them designs, or  $t$ -designs when the degree of regularity is to be emphasized. A design can be defined formally as follows.

**Definition 3.4.2** *An incidence structure  $\mathcal{D} = (\mathcal{P}, \mathcal{B}, \mathcal{I})$  is a  $t$ -design or a  $t - (v, k, \lambda)$  design (where  $v, k, \lambda$  are non-negative integers called parameters) is an incidence structure  $\mathcal{D} = (\mathcal{P}, \mathcal{B}, \mathcal{I})$  such that  $|\mathcal{P}| = v$ ,  $\beta \in \mathcal{B}$  is incident with  $k$  points and  $t$  distinct points are together incident with  $\lambda$  blocks. Symmetric design is a design with the same number of points and blocks.*

A  $2 - (v, k, \lambda)$  design is called a block design. We say that a  $t - (v, k, \lambda)$  design  $D$  is a quasi-symmetric design with intersection numbers  $x$  and  $y$  ( $x < y$ ) if any two blocks of  $D$  intersect in either  $x$  or  $y$  points (see[16, 17]).

Example 3.4.1 is a  $t$ -design provided that the cardinality of subset  $\mathcal{B}$  of  $\mathcal{P}$  is  $k$  for every subset  $\beta \in \mathcal{B}$  and that for every subset  $\mathcal{T}$  of  $\mathcal{P}$  of cardinality  $t$ ,  $|\{\beta \in \mathcal{B} | \beta \supseteq \mathcal{T}\}| = \lambda$ . Thus all blocks have the same cardinality and every  $t$ -subset is contained in the same number of blocks.

**Example 3.4.2** Let  $\mathcal{P} = \{1, 2, 3, 4, 5, 6, 7\}$  and set

$$\mathcal{B} = \{\{1, 2, 3\}, \{1, 4, 5\}, \{1, 6, 7\}, \{2, 4, 7\}, \{2, 5, 6\}, \{3, 5, 7\}, \{3, 4, 6\}\}.$$

Then  $(\mathcal{P}, \mathcal{B})$  with a natural incidence forms a  $3 - (7, 3, 1)$  design.

**Remark 3.4.1** A  $t - (v, k, \lambda)$  design is also referred to as a  $t$ -design. We shall assume that all parameters are positive integers and  $v > k \geq t$  to avoid trivial cases. Also members of  $\mathcal{B}$  are distinct, thus there are no repeated blocks.

**Theorem 3.4.2** Let  $P = (\mathcal{P}, \mathcal{B})$  be a  $t - (v, k, \lambda)$  design. Then for every integer  $s$  such that  $0 \leq s < t$ , the number  $\lambda_s$  of blocks incident with  $s$  distinct points is independent of the  $s$  points and is given by

$$\lambda_s = \lambda \frac{(v-s)(v-s-1)\dots(v-t+1)}{(k-s)(k-s-1)\dots(k-t+1)}.$$

In particular  $\mathcal{D}$  is an  $s - (v, k, \lambda_s)$  design for every  $s$  with  $1 \leq s \leq t$ .

**Proof:** Suppose that  $X \subset \mathcal{P}$  with  $|X| = s$ . Let  $\lambda_s(X)$  denote the number of blocks containing all the points in  $X$ . Define the set

$$\mathcal{T} = \{(Y, B) | Y \subset \mathcal{P}, B \in \mathcal{B}, |Y| = t - s, Y \cap X = \emptyset, Y \cup X \subset B\}.$$

We compute  $|\mathcal{T}|$  in two different ways.

First there are  $\binom{v-s}{t-s}$  ways to choose  $Y$ . For each  $Y$ , there  $\lambda$  blocks  $B$  such that  $X \cup Y \subset B$ . Hence

$$|\mathcal{T}| = \lambda \binom{v-s}{t-s}.$$

On the other hand, there are  $\lambda_s(X)$  ways to choose a block  $B$  such that  $X \subset B$  for each choice of  $B$  there are  $\binom{k-s}{t-s}$  ways to choose  $Y$ . Hence

$$|\mathcal{T}| = \lambda_s(X) \binom{k-s}{t-s}.$$

Combining two equation we see that  $\lambda_s(X) = \lambda_s$  and the results follows. □

**Definition 3.4.3** Suppose  $D = (P, \beta, I)$  is a design in which  $P = \{p_1, p_2, \dots, p_v\}$  and  $B = \{B_1, B_2, \dots, B_i\}$ . Then the incidence matrix of  $D$  is a  $b \times v$  matrix  $A = (a_{ij})$  such that

$$a_{ij} = \begin{cases} 1, & \text{if } (p_i, B_j) \in I \\ 0, & \text{if } (p_i, B_j) \notin I \end{cases}$$

The incidence matrix depends on the ordering of points and blocks. If we impose the labelling on the points of a design  $\mathcal{D}$ ,  $\{p_1, p_2, p_3, \dots, p_v\}$ , block  $\mathcal{B}$  of the design can be represented as an incidence vector  $v^B$  of length  $v$  (see[7]).

**Definition 3.4.4** A Steiner system is a type of block design, specifically a  $t -$  design with  $\lambda = 1$  and  $t \geq 2$ .

A Steiner system with parameters  $t, k, n$  written  $S(t, k, n)$  for  $n > 1$  is an  $n$ -element set  $S$  together with a set of  $k$ -element subsets of  $S$  (called blocks) with the property that each  $t$ -element subset of  $S$  is contained in exactly one block.

### 3.4.1 The construction of $t$ -designs from linear codes

Let  $C$  be a linear code of length  $n$ . Consider all the code words of weight  $w$  in  $C$ . Let  $c = (c_1, c_2, c_2, \dots, c_n)$  be a codeword of weight  $w$  in  $C$ . The support of  $c$  is defined by

$$\text{suppt}(c) = \{1 \leq i \leq n : c_i \neq 0\} \subseteq \{1, 2, 3, \dots, n\}.$$

Two different codewords of weight  $w$  may have the same support. Let  $\mathcal{P} = \{1, 2, 3, \dots, n\}$  and  $\mathcal{B}$  be the set of the supports of the codewords of weight  $w$  in  $C$ , where no repeated blocks are allowed. Let the incidence relation  $\mathcal{R}$  be the usual containment of sets. Then it is possible that  $(\mathcal{P}, \mathcal{B}, \mathcal{R})$  is a  $t$ -design for some  $t$ . In this case, we say that the codewords of weight  $w$  in  $C$  hold or support a  $t$ -design, which is called a support design of  $C$ .

The Assmus-Mattson Theorem describes  $t$ -designs from linear codes



**Theorem 3.4.3 (Assmus-Watson)** *Let  $C$  be an  $[n, k, d]$  code over  $GF(q)$ . Let  $d^\perp$  denote the minimum distance of  $C^\perp$ . Let  $w$  be the largest integer satisfying  $w \leq n$  and*

$$w - \lfloor \frac{w + q - 2}{q - 1} \rfloor \leq d.$$

*Where  $\lfloor x \rfloor$  is the largest integer less than or equal to  $x$ . Define  $w^\perp$  analogously using  $d^\perp$ . Let  $(A_1)_{i=0}^n$  and  $(A_1^\perp)_{i=0}^n$  denote the weight distribution of  $C$  and  $C^\perp$ , respectively. Fix a positive integer  $t$  with  $t < d$ , and let  $s$  be the number of  $i$  with  $A_1^\perp \neq 0$  for  $1 \leq i \leq n - t$ . Suppose  $s \leq d - 1$ . Then*

- i. the codewords of weight in  $C$  hold a  $t$ -design provided  $A_1 \neq 0$  and  $d \leq i \leq w$ , and*
- ii. the codewords of weight in  $C^\perp$  hold a  $t$ -design provided  $A_1^\perp \neq 0$  and*

$$d^\perp \leq i \leq \min\{n - t, w^\perp\}$$

**Proof:** See [20, Theorem 4.24]. □

The Assmus-Mattson Theorem applied to  $C$  is most useful when  $C^\perp$  has only a few nonzero weights. It has been one of the two tools used for computing designs in linear codes. When  $q = 2$ , Theorem 3.4.3 becomes the following.

**Corollary 3.4.4 (Assmus-Watson)** *Let  $C$  be an  $[n, k, d]$  code over  $GF(q)$ . Let  $d^\perp$  denote the minimum distance of  $C^\perp$ . Let  $(A_1)_{i=0}^n$  and  $(A_1^\perp)_{i=0}^n$  denote the weight distribution of  $C$  and  $C^\perp$ , respectively. Fix a positive integer  $t$  with  $t < d$ , and let  $s$  be the number of  $i$  with  $A_1^\perp \neq 0$  for  $1 \leq i \leq n - t$ . Suppose  $s \leq d - t$ . Then*

- i. the codewords of weight in  $C$  hold a  $t$ -design provided  $A_1 \neq 0$  and  $d \leq i \leq n$ , and*
- ii. the codewords of weight in  $C^\perp$  hold a  $t$ -design provided  $A_1^\perp \neq 0$  and  $d^\perp \leq i \leq n - t$ .*

**Proof:** See [20, Corollary 4.26]. □

**Remark 3.4.5** *If a primitive permutation automorphism group  $G$  of degree  $n$  is contained in the permutation automorphism group of a code  $C$ , then we say that  $C$  admits  $G$  as a primitive permutation automorphism group.*

Note that  $A = PAut(C)$  is also of degree  $n$  and all codewords in  $C$  are of length  $n$ . If  $g \in A$ , then for  $1 \leq i, j \leq n$  we have  $i^g = j$  if and only if for any codeword  $v \in C$ , the  $i^{th}$  coordinate of  $v^g$  is replaced by  $j^{th}$ . This is how  $A$  acts on  $C$ . For a positive integer  $m$ , we define:  $W_m(C) = \{v \in C : wt(v) = m\}$ . If there is no ambiguity, we may simply write  $W_m$ . Since the automorphisms of  $C$  preserve the weight of codewords, we deduce that  $A$  acts on  $W_m$  for every integer  $m$  with  $W_m \neq \emptyset$ . The stabilizer of this action is of interest. If  $v \in W_m$ , then the stabilizer of  $v$  in  $A$  is the set of all  $g \in A$  with  $vg = v$ . So if the code is binary, the stabilizer of  $v$  in  $A$  is isomorphic to the stabilizer of the support of  $v$  in  $A$ . We can see that some 1-designs may be constructed from the codes, using this action.

**Proposition 3.4.6** *Let  $C = [n, k, d]_2$  be a binary linear code admitting  $G$  as permutation automorphism group and  $W_m(C) \neq \emptyset$ . If  $S$  is an orbit of the action of  $G$  on  $W_m$ , then we have a  $1 - (n, m, m|S|/n)$  design with block set  $B = \{Supp(w) : w \in S\}$ .*

**Proof:** See [19, Proposition 1]. □

**Corollary 3.4.7** *Let  $C = [n, k, d]_2$  be a binary linear code admitting  $G$  as permutation automorphism group and  $W_m(C) \neq \emptyset$ . If  $|S| = n$ , then the resulting support 1-design is a symmetric design.*

**Proof:** The results follows directly from Proposition 3.4.6, if we have  $|S| = n$  then  $1 - (n, m, m|S|/n) = 1 - (n, m, m)$  hence the symmetric design. □

We say that an incidence structure  $I$  is transitive if an automorphism group of  $I$  acts transitively on points and blocks. An incidence structure  $I$  is called primitive if an automorphism group acts primitively on points and blocks (see [26, 27]). In our dissertation we will use construction method described in Theorem 3.4.6 to construct 1-designs.

# Chapter 4

## Constructions of combinatorial structures

In this chapter, we turn our attention to the methods used in this dissertation for the construction of codes and designs. These methods are the core of our research, enabling us to develop algorithms that uncover structures and properties within coding theory and design theory.

Central to these methodologies is the utilization of MAGMA [10], a powerful computational algebra system. MAGMA plays a pivotal role in our work, providing the computational requirements to implement and test the algorithms that underlie our constructions. Other computational algebra system such as GAP [22] can be used, but for our purpose they are very limited compared to MAGMA.

### 4.1 $\mathbb{F}G$ -modules and $G$ -invariant codes

#### 4.1.1 Codes from quotient modules

Constructing codes from quotient modules is a common and important technique in coding theory. Quotient modules, also known as factor modules, are derived by dividing a module by one of its submodules.

Given a representation of group elements of a group  $G$  by permutations one can work modulo  $q$  and obtain a representation of  $G$  on a vector space  $V$  over  $\mathbb{F}_q$ . The invariant subspaces are then all the binary codes  $C$  for which  $G$  is a subgroup of  $Aut(C)$ . In the context of coding theory, this approach involves creating codes based on the quotient of a vector space  $V$  by one of its subspaces  $W$ . The resulting quotient space, denoted as  $V/W$ , forms the basis for constructing specific types of codes. (see [14])

If we consider a vector space  $V$  over a finite field  $\mathbb{F}_q$  and a subspace  $W$  of  $V$ . The quotient space  $V/W$  consists of cosets of  $W$  in  $V$ . Each coset represents an equivalence class of vectors that have the same remainder when divided by the vectors in  $W$ . This quotient space inherits a vector space structure from  $V$ , enabling the creation of codes based on its elements. The properties of such codes, such as minimum distance and error correcting capabilities depends on our choice of  $W$ . So careful selection of  $W$  is crucial in this method.

### 4.1.2 Codes from maximal submodules

In this dissertation we are interested in  $G$ -invariant codes from the primitive permutation representations, hence we shall consider the permutation modules obtained from the action of the group on the coset of its maximal subgroups and thus determine the corresponding  $\mathbb{F}\Omega$ -submodules, in particular maximal submodules.

Given a permutation group  $G$  on a finite set  $\Omega$  and a finite field  $\mathbb{F}$  (i.e. the vector space over  $\mathbb{F}$  with basis  $\Omega$ ). The  $G$ -invariant submodules can be regarded as linear codes (see Theorem 3.2.1), and therefore we may find the properties of these codes such as weight distributions. In this dissertation we use the approach of submodules, which involves the study of substructures within modules, which are algebraic structures closely related to vector spaces. Thus we determine all binary codes invariant under a given group more directly, since we obtain explicit bases for the codes. Moreover, for each primitive representation of a given permutation group  $G$ , we use MAGMA to construct the associated permutation module over  $\mathbb{F}_2$  and find all maximal submodules. These submodules are the  $G$ -invariant codes and we also find

lattices of submodules.

Let  $G$  be a finite primitive group action on set  $X$ ,  $H = G_\alpha$  its maximal subgroup, where  $\alpha \in X$ . Consider the action of  $G$  on a set of cosets  $\Omega = \{G, G/H\}$ , where  $G/H = \{gH : g \in G\}$  (see [14]). According to Theorem 1.2.1 and Theorem 1.2.2,  $G$  acts transitively and primitively on  $\Omega$  and its image is a permutation representation. From this we are able to construct  $\mathbb{F}\Omega$ -permutation modules over  $\mathbb{F}_q$  corresponding to this representation. We shall consider these permutations to construct subspaces (i.e submodules). The  $G$ -invariant subspaces (i.e., submodules) of the permutation module give all the  $p$ -ary codes invariant under  $G$ . The approach offered by this section, which is at the core of the purpose of the dissertation, is more inclusive than those presented in Section 4.1.1. The codes constructed using that method are in general subcodes of the ones constructed using those method that we present in the ensuing section. Since this dissertation is concerned with binary codes we focus mainly on the field  $\mathbb{F} = \mathbb{F}_2$ . So the step by step procedure to find these codes start with the vector space  $V$  over a field  $\mathbb{F}_q$ , this is the fundamental space in which we want to construct our codes. Within the vector space we identify the submodules, the submodule are the subspaces of the vector which are also vector spaces, and they are essential in creating our codes. From those submodules then we focus on those which are maximal. Codes are constructed based on the properties of these maximal submodules. Specifically, you use the maximal submodules to define the structure and properties of your code. The codes derived from maximal submodules are designed to have specific error-correcting capabilities and performance characteristics.

### 4.1.3 Permutation codes

Permutation codes is a group of error correcting codes based on a group action of permutation group. These codes have unique properties and are good for applications where the groups action on the codewords is relevant. These codes are constructed by the action of a permutation group on a set. On constructing these codes we first determine maximal subgroups of the desired group. These subgroups are essential for constructing permutation

codes, we then find set of conjugates of each subgroup. These are found applying elements of the group to the subgroup using the conjugation operation. Noting that each set of conjugates of these maximal subgroups represents a codeword and they are typically binary vectors or over elements from a finite field. we then use action by conjugates to transform each codeword into another codeword and this action is important for the construction of permutation codes. we then use the orbits of the point stabilizer to generate a permutation module over a suitable field such as  $\mathbb{F}_2$  or  $\mathbb{F}_3$  and they represent how the  $G$  permutes the conjugates of its maximal subgroups and they are the basis of our codes. The permutation code is constructed by considering the codewords in the permutation module. The codewords are determined based on the orbits and stabilizers. Each codeword is a combination of elements from the set of conjugates.

## 4.2 Construction of $G$ -invariant codes.

The construction used in this dissertation is based on method described in section 4.1, Theorem 3.2.1, Theorem 4.2.1 and Theorem 4.2.2, assuming that the Schur multiplier of the group is trivial and that the codes used are binary. If the codes are non-binary, it might be the challenge to identify all the codes accurately. However, even in such cases, our methods can still be utilized under the condition that  $(|M/M'|, |\mathbb{F}_n^*|) = 1$ . This condition provides exception where the methods remain applicable despite the field being non-binary.

**Definition 4.2.1** *Let  $G$  be a finite group, a Schur cover or a covering group or stem cover of  $G$  is a finite group  $H$  with a normal subgroup  $N \subseteq Z(H) \cup H$ , with  $H/N \cong G$ , that has maximal order amongst all groups with this property.*

**Lemma 4.2.1** *Assume  $(G, X)$  is a primitive permutation group and  $\mathbb{F}$  a field such that  $\text{Ext}(G/G', \mathbb{F}^*) = 0$ . Let  $E$  be a stem cover of  $G$  and  $E_0$  the inverse in  $E$  of the stabilizer of  $G$  induced up to  $E$  all 1 – dimensional  $\mathbb{F}E_0$ -module. Then the submodules of the resulting  $\mathbb{F}E$ -modules provide for a complete list of codes over  $\mathbb{F}$  admitting  $(G, X)$  as a permutation group.*

**Proof:** See [19, Theorem 1]. □

**Theorem 4.2.2** *Let  $G$  be a finite simple group with a maximal subgroup  $M$ . Let  $P$  be the permutation  $\mathbb{F}_n G$ -module corresponding to the primitive action of  $G$  on  $M$ , where  $\mathbb{F}_n$  is a finite field. Also assume that the Schur multiplier of  $G$  is trivial and  $(|M/M'|, |\mathbb{F}_n^*|) = 1$ , for  $M'$  a derived group of  $M$ . Then the set of linear codes of length  $m$  over  $GV(q)$  equals the set of all submodules of  $P$ . (see [19, Theorem 2])*

**Proof:** Since the Schur multiplier of  $G$  is trivial. Then  $G$  is its own covering group. Therefore by Lemma 4.2.1 the set of linear codes of length  $m$  over  $GV(q)$  is the set of all submodules of the induced modules of 1 – dimensional  $\mathbb{F}M$ -modules of  $G$ . Let  $\rho$  be a representation of  $M$  of degree 1. We claim that  $\rho$  is trivial. Indeed  $M/Ker\rho$  lies in a group over  $\mathbb{F}^*$ . As  $\mathbb{F}_n^*$  is abelian we have  $M' \leq Ker(\rho)$ . Hence  $|M : Ker(\rho)|$  divides both  $|\mathbb{F}_n^*|$  and  $|M/\mathbb{F}_n^*|$ . Hence we have  $M = ker(\rho)$  and the result follows. We conclude that set of linear codes of length  $m$  over  $GLV(q)$  is a set of all submodules of the permutation module of  $G$  of degree  $|G : M|$ , hence the result follows. □

Given a permutation module acting on a set  $\Omega$ , and  $\rho : G \rightarrow GF(V)$  where  $\rho(g)v = g.v$  for  $g \in G$  and  $v \in V$  we can find all codes with a group  $G$  acting as an automorphism group as follows:

1. Let  $\mathbb{F}\Omega$  be a permutation module.
2. Use MAGMA to find all  $\mathbb{F}\Omega$  submodules.
3. By Theorem 3.2.1 and Theorem 4.2.2 these submodules are  $G$ -invariant codes.

The construction therefore enables us to find all submodules of the permutation module. For this we decompose the permutation module into submodules. These constitutes the building blocks for the construction of a lattice of submodules where possible. With the characterization of these codes we respond to the problem of classification of the codes. As was discussed in Chapter 3.2 decomposition of the modules into submodules depends on the field. Maschke’s theorem (see Theorem 2.3.1) gives a characterization of decomposition over

a field whose characteristic is 0 or relatively prime to the order of the group. In this case the permutation module is completely reducible and can be written as a direct sum of its irreducible submodules. When a characteristic  $p$  of a field divides the order of the group i.e.,  $p \mid |G|$ , we apply Krull-Schmidt's theorem (see Theorem 2.4.2) which shows that any module with finite length can be written as a direct sum of indecomposable submodules, and this decomposition is unique up to isomorphism and the order of the summands. In addition to Krull-Schmidt Theorem, we have the composition series of the module which provides a way of breaking the module into simple components. These concepts have been used to develop different methods to construct submodules hence codes invariant under a group. Applying all these theories and techniques we construct  $G$ -invariant codes which have certain classes of finite simple groups acting irreducibly on. In Chapter 5 we shall see how the techniques outlined in Section 4.1 and Section 4.2 help us determine and classify a number of interesting codes invariant under the simple group  $U(3, 3)$ .

**Corollary 4.2.3** *A binary code is even if and only if it is contained in the dual of a repetition code. [19, Corollary 1]*

**Proof:** Let  $C_1$  be a repetition code Assume that a binary code  $C \subseteq C_1^\perp$ .  $C^\perp$  consists of all binary sequences of a fixed length  $n$  where each element is either 0 or 1. This means  $C^\perp$  contains all possible binary words of length  $n$ . If code  $C \subseteq C^\perp$ , it means that each codeword in  $C$  can be expressed as a linear combination of the basis vectors of the  $C^\perp$ , which are the binary words of length  $n$ . Since each codeword in  $C$  is a linear combination of binary words of length  $n$ , it means that the weight in each codeword of  $C$  is even. This is because each binary word in the  $C^\perp$  has an even weight, and a linear combination of binary words with even weights will also have an even weight. Therefore, if  $C$  is even.

*Conversely:* Assume that a binary code  $C$  is even, meaning that all codewords in  $C$  have even weights. We can construct a repetition code by taking a binary word of length  $n$  and repeating it multiple times to form a code. Let's say the repeated word is  $0^n$  (a binary word of  $n$  zeros). Now, each codeword in this repetition code is a repetition of  $0^n$ . Since all codewords in  $C$  have even weights, they can be expressed as linear combinations of  $0^n$  (with



suitable coefficients). This means that  $C$  is contained in the dual of this repetition code.  $\square$

**Lemma 4.2.4** *Let  $C$  be a code of length  $n$  over a finite field  $\mathbb{F}_q$  and  $C^\perp$  the dual of code  $C$ . The all-one codeword  $1$  lies in  $C$  if and only if  $q$  divides the sum of all coordinates of each  $w \in C^\perp$ . In particular if  $q = 2$ , then  $1 \in C$  if and only if the length is even.*

**Proof:** If  $1 \in C$ , then for all  $w \in C^\perp$  we have  $\langle w, 1 \rangle = 0$ . Since  $\langle w, 1 \rangle$  is the sum of all coordinates of  $w$ , we get  $q | \langle w, 1 \rangle$ . This completes the proof.  $\square$

# Chapter 5

## Codes invariant under $U(3,3)$

In this chapter we consider the unitary group  $G = U(3, 3)$  a classical simple group of order 6048 of Lie type with Steiner system  $S - (2, 4, 28)$  (see[11]) acting 2-transitive in 28 points. The Schur multiplier of this group is trivial. We then use the discussed methods to construct codes invariant under this group and also construct designs from supports of those codes.

### 5.1 The structure of the unitary group $U(3,3)$

According to Atlas (see[43])  $U(3, 3)$  has 4 primitive permutation representation listed in Table 5.1 where second column shows the structure of maximal subgroup and the third column shows the degree which is number of cosets of point stabilizer.

Table 5.1: Maximal subgroups of  $U(3, 3)$

Number	Maximal subgroup	degree
$M_1$	$3_+^{1+2} : 8$	28
$M_2$	$L(2, 7)$	36
$M_3$	$4.S_4$	63
$M_4$	$4^2 : S_3$	63

The table shows that the group has 4 class maximal subgroups up to conjugation. So

for each maximal subgroup  $M_k$ ,  $k = 1, 2, 3, 4$ , the action of  $G$  by conjugation on the set of conjugates of  $M_k$  giving us the primitive action of degree  $|G:M_k| \in \{28, 36, 63\}$ . The elements of each degree generate a permutation module over  $\mathbb{F}_2$ . We determine orbits of the point stabilizer through coset action of  $G$  on the maximal subgroups. According to Theorem 4.2.2 every binary code of length  $|G:M_k|$  that admits  $G$  as a primitive permutation group is a submodule of the permutation module of  $G$  with respect to the action of  $G$ . Moreover if  $(|M_k:M'_k|, q - 1) = 1$  then the same results holds for codes over  $\text{GF}(q)$ . For example if  $M_k \cong L(2, 7)$  then  $M_k$  is perfect therefore we can find linear codes of length 36 in  $\text{GF}(q)$  where  $q$  divides the order of  $G$ . However, for the rest of the maximal subgroups  $M_k$  of  $G$ , the value of  $|M_k:M'_k|$  happens to be 2, 4 or 8 so it suffices to find only binary linear codes of given length using our method. Using this method, we can only compute all binary codes of given length. Using ATLAS or MAGMA, we have that the unitary group  $U(3, 3)$  has 5 irreducible  $G$ -modules in  $\text{GF}(2)$  with dimensions 1, 6, 14, 32, 32 and they are all absolutely irreducible.

Our aim is to find the whole codes of length  $|G:M_k|$  that contain  $G$  in their permutation automorphism group. Denote by  $P_i(q)$  the permutation module over  $\text{GF}(q)$  with respect to the primitive action of  $G = U(3, 3)$  on the set of the conjugates of  $M_k$  in  $G$ . We want to find the set of all binary codes whose automorphism groups contain  $G$ . According to Theorem 4.2.2, the codes are of type  $c = [n, d, m]_2$ , where  $n \in \{28, 36, 63\}$ . If  $d = 0$  or  $d = n$ , then we consider the code as trivial. So the trivial codes are of type  $c = [n, 0, m]_2$  and  $c = [n, n, m]$ . Also if  $d = 1$  the  $C$  is a repetition code therefore repetition codes are of type  $C = [n, 1, m]$ . As the results of the following lemma we can see that the repetition code and its dual have restricted structure.

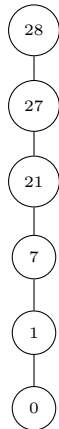
According to Theorem 4.2.1 the codes  $[n, n, 1]$  and  $[n, n - 1, 2]$  corrects less than 1 errors, so are of less interest.

## 5.2 Permutation representations and permutation modules of $U(3,3)$

### 5.2.1 Representation of degree 28

For a permutation group acting on set  $\Omega$  of degree 28, we find a 28-dimensional permutation module invariant under  $G$ . We take the permutation module to be our working module and find all submodules. That permutation module breaks into submodules of dimension 0, 1, 7, 21 and 27, with composition series  $V = 28 \supset 27 \supset 21 \supset 7 \supset 1 \supset 0$ . The submodule lattice of this representation is shown on Figure 5.2.1. This shows that the only irreducible submodule is of dimension 1. The non-trivial submodules are dimension 1, 7, 21 and 27. Therefore according to Theorem 4.2.2 it suffices to find codes of dimension 1, 7, 21 and 27 and we will denote these codes by  $C_{28,1}$ ,  $C_{28,2}$ ,  $C_{28,3}$  and  $C_{28,4}$  respectively. Codes arising

Figure 5.1: Submodule lattice for a 28-dimensional representation



from the submodule lattice in Figure 5.2.1 are shown in Table 5.2.

**Proposition 5.2.1**  $C_{28,1}$  is a repetition code and  $C_{28,4}$  is its dual code and they have minimum distances 28 and 2 respectively. Moreover  $C_{28,1} \leq C_{28,4}$ , and  $C_{28,1}$  is irreducible.

**Proof:** The dimension of  $C_{21,1}$  satisfies the definition of repetition code. Moreover  $|P_1(2)| - 1 = 27$  by Proposition 2.4.8 it follows that  $C_{28,4}$  is a dual code of  $C_{28,1}$ . Now by Lemma

Table 5.2: Codes of length 28

Naming	Code	Weight distribution
$C_{28,1}$	$[28, 1, 28]$	$28^1$
$C_{28,2}$	$[28, 7, 12]$	$0^1, 12^{63}, 16^{63}, 28^1$
$C_{28,3} = C_{28,2}^\perp$	$[28, 21, 4]$	$0^1, 4^{315}, 6^{6048}, 8^{47817}, 10^{206976}, 12^{472059}, 14^{630720},$ $16^{472059}, 18^{206976}, 20^{47817}, 22^{6048}, 24^{315}, 28^1$
$C_{28,4} = C_{28,21}^\perp$	$[28, 27, 2]$	$0^1, 2^{378}, 4^{20475}, 6^{376740}, 8^{3108105}, 10^{13123110}, 12^{30421755}, 14^{40116600},$ $16^{30421755}, 18^{1312311}, 20^{3108105}, 22^{376740}, 24^{20475}, 26^{378}, 28^1$

3.2.8  $C_{21,1} = [28, 1, 28]$  and  $C_{28,4} = [28, 27, 2]$  hence  $C_{28,4}$  is the dual code of  $C_{21,1}$ .  $\text{Char}(\mathbb{F}_2)$  divides the order of  $|P_1(2)|$  therefore also by Proposition 2.4.8 we have  $C_{21,1} \leq C_{28,4}$  as proposed.  $\square$

Using MAGMA we find that minimum distance of  $C_{28,2}$  is 12 and minimum distance of  $C_{28,3}$  is 4. From the fact that if  $C = [n, k, d]$  then  $C^\perp = [n, n - k, d']$  we can conclude that  $C_{28,2}$  and  $C_{28,3}$  are each other's dual code. From definition of trivial codes we have the code of form  $C_{28,0} = [28, 0, 28]$  and  $C_{28,5} = [28, 28, 1]$  and the code  $C = [28, 1, 28]$  satisfies properties of repetition code.

**Proposition 5.2.2** *Let  $M$  be a primitive subgroup of degree 28 of the unitary group  $U(3, 3)$  and  $C_{28,2}$  a binary linear code of dimension 7. Then  $C_{28,2}$  is doubly even.*

**Proof:** Using MAGMA the weight distribution of  $C_{28,2}$  is

$$\{0^1, 12^{63}, 16^{63}, 28^1\}$$

since all the weights of this code are divisible by 4,  $C_{28,2}$  is doubly even.  $\square$

**Proposition 5.2.3** *Let  $G$  be a primitive group of degree 28 of the unitary group  $U(3, 3)$  and  $C_{28,3}$  a binary linear code of dimension 21. Then  $C_{28,3} = [28, 21, 4]$  is even but not doubly even and it is self-orthogonal.*

**Proof:** The weight distribution of  $C_{28,3}$  is

$$\{0^1, 4^{315}, 6^{6048}, 8^{47817}, 10^{206976}, 12^{472059}, 14^{630720}, 16^{472059}, 18^{206976}, 20^{47817}, 22^{6048}, 24^{315}, 28^1\}$$

all the weights of this code are even therefore it is even, but 6 is not divisible by 4 therefore the code is not doubly even. Furthermore the dual of  $C_{28,3}$  is  $C_{28,2}$  and according to the composition series it is its subspace therefore  $C_{28,3}$  contains its dual therefore it is self orthogonal.  $\square$

Since  $C_{28,3}$  is binary, it follows from Corollary 4.2.3 that  $C_{28,3} \leq C_{28,4}$ .

**Proposition 5.2.4** *every 28 dimensional binary code accepting  $U(3, 3)$  as primitive permutation group is even.*

**Proof:** Using the composition series we note that all the non-trivial submodules are contained in the 27 dimensional submodule, this implies all the codes are contained in the dimensional code which is the dual of the repetition code therefore by Corollary 4.2.3 all the codes are even.  $\square$

**Proposition 5.2.5** *The non-trivial codes  $C_{28,2} = [28, 7, 12]$  and  $C_{28,3} = [28, 21, 4]$  can correct up to 5 and 1 errors respectively.*

**Proof:** Since the minimum weight of  $C_{28,2} = 12$  and  $C_{28,3} = 4$  then by Theorem 3.3.2 a code can correct up to  $\frac{d-1}{2}$  errors and the results follows.  $\square$

Table 5.3 shows some of the properties of the non-trivial codes we constructed. The computations were based on MAGMA.

Table 5.3: non-trivial codes of length 28

Code	Parameters	$\text{Aut}(C_i)$	Primitivity of $C_i$
$C_{28,2}$	[28, 7, 12]	$S(2, 6)$	yes
$C_{28,3}$	[28, 21, 4]	$S(2, 6)$	yes

### Designs held by support of codewords in $C_{28,i}$

Suppose that  $w_n$  is a codeword of nonzero weight  $m$  in a non-trivial code  $C = C_{28,i}$  where  $i = 1, 2, 3, 4$ . In this section we determine the structure of  $(Aut(C))_{w_m}$ , that is the stabilizer of  $w_m$  in  $Aut(C)$  where  $w_m = \{c \in C : w(c) = m\}$ .

**Definition 5.2.1** *The designs constructed from the supports of codes are called support designs.*

We now examine the action of  $Aut(C_i)$  on the set of  $w_m$  of non-trivial codewords of  $C$  and describe their nature. In addition we look at the structure of stabilizer  $(Aut(C))_{w_m}$  and construct the support 1-design using Theorem 3.4.6 and furthermore we take the image of the support of  $w_m$  under the action of  $G = Aut(C)$  to find the blocks of the  $2 - (n, m, k_m)$  design where  $k_m = |(w_m)^G| \times \frac{m}{n}$  and show that  $Aut(C)$  acts primitively on those designs.

**Proposition 5.2.6** *Let  $w$  be a codeword of the code  $C_{28,i}$  ( $i = 2, 3$ ) of weight  $m$  and  $A = Aut(C_i)$  if  $w_m \neq \emptyset$  then the action of  $A$  on  $W_m(C_i)$  is transitive. The stabilizer of  $w = w_m(C_i)$  in  $A$  is a maximal subgroup of  $U(3, 3)$  and the support designs constructed from these codes are shown in Table 5.4 and Table 5.5.*

Table 5.4: Stabilizer and support designs from [28, 7, 12]

<b>m</b>	$s =  w_m $	<b>Stabilizer</b>	<b>Maximal in <math>A</math></b>	<b>Design</b>
12	63	$2^5 : S_6$	yes	$1 - (28, 12, 27)$
16	63	$2^5 : S_6$	yes	$1 - (28, 16, 36)$

Table 5.5: Stabilizer and support designs from [28, 21, 4]

<b>m</b>	<b>Orbits' size</b>	<b>Stabilizer</b>	<b>Maximal in <math>A</math></b>	<b>Design</b>
4	315	$2.[2^6] : (S_3 \times S_3)$	yes	$1 - (28, 4, 45)$

6	1008	$A_6.2^2$	no	$1 - (28, 6, 216)$
	5040	$A_6.2^2$	no	$1 - (28, 6, 1080)$
8	945	$(4^2 \times 2).2^3.S_3$	no	$1 - (28, 8, 215)$
	22680	$(4^2 \times 2).2^3.S_3$	no	$1 - (28, 8, 6480)$
	24192	$(4^2 \times 2).2^3.S_3$	no	$1 - (28, 8, 6912)$
18	336	$S_3 \times S_6$	yes	$1 - (28, 18, 216)$
	5040	$S_3 \times S_6$	yes	$1 - (28, 18, 3240)$
	15120	$S_3 \times S_6$	yes	$1 - (28, 18, 9720)$
22	1008	$A_6.2^2$	no	$1 - (28, 22, 792)$
	5040	$A_6.2^2$	no	$1 - (28, 22, 3690)$
24	315	$2.[2^6] : (S_3 \times S_3)$	yes	$1 - (28, 24, 270)$

**Proof:** The values of  $m$  and  $s$  are given in the weight distribution of the codes. Using Theorem 3.4.6 the designs from the supports of these codes are of the form  $1 - (n, m, \lambda)$  where  $n$  is the length of the code and  $\lambda = ms/n$ , where  $s$  is the size of orbits of  $w_m$ . Transitivity, stabilizers and maximality of stabilizer in  $A$  was determined using MAGMA and the results follows.  $\square$

We notice that number of designs rely on number of orbits of  $w_m$  an orbits' size is the number of blocks in the design. All support designs under each  $m_i$  have the same stabilizer so we can add all orbits' size and get  $|w_m|$  and use it to find one design with large parameter  $\lambda$ , for example with  $m_2 = 6$  we can have  $|w_m| = 1008 + 5040 = 6048$  and construct a design  $1 - (28, 6, 1296)$  with 6048 blocks and stabilizer  $A_6.2^2$ .

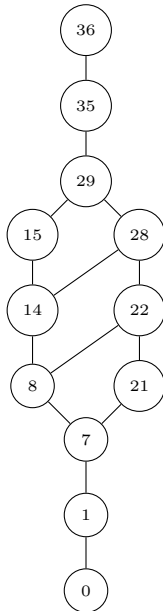
## 5.2.2 Representation of degree 36

We construct a 36-dimensional permutation module invariant under permutation group  $G = U(3, 3)$  acting on set  $\Omega$  of degree 36. We let the permutation module be our working module and recursively find all submodules. We find that permutation module breaks into 11 submodules of dimension 0, 1, 7, 8, 14, 15, 21, 22, 28, 29 and 35. The submodules are



the dimensions of the codes related to permutation module. The submodule lattice of this representation is shown in Figure 5.2.2, showing that only the code of dimension one is irreducible.

Figure 5.2: Submodule lattice for a 36-dimensional representation



According to Theorem 4.2.2 there exist codes of length 36 and dimension  $k$  where  $k \in \{0, 1, 7, 8, 14, 15, 21, 22, 28, 29, 35\}$ . We already know by definition of trivial codes that we have  $C_{36,1} = [36, 0, 36]$ ,  $C_{36,2} = [36, 1, 36]$ ,  $C_{36,11} = [36, 35, 2]$  and  $C_{36,12} = [36, 36, 1]$ . What is left now is find the minimum weight of remaining non-trivial codes.

**Proposition 5.2.7** *Let  $C$  be a non-trivial binary linear code of length 36 which is invariant under  $U(3, 3)$ . Then  $C$  is one of the following codes.*

1.  $C_{36,2} = [36, 1, 36]$ .
2.  $C_{36,3} = [36, 7, 16]$ .
3.  $C_{36,4} = [36, 8, 14]$ .
4.  $C_{36,5} = [36, 14, 8]$ .
5.  $C_{36,6} = [36, 15, 8]$ .
6.  $C_{36,7} = C_{36,6}^\perp = [36, 21, 6]$ .
7.  $C_{36,8} = C_{36,5}^\perp = [36, 22, 6]$ .
8.  $C_{36,9} = C_{36,4}^\perp = [36, 28, 4]$ .

$$9. C_{36,10} = C_{36,3}^\perp = [36, 29, 4].$$

$$10. C_{36,11} = C_{36,2}^\perp = [36, 35, 2].$$

**Proof:** Using MAGMA we see that  $P_2(2)$  contains 10 non-trivial proper submodules of degree of dimensions;

$$1, 7, 8, 14, 15, 21, 22, 28, 29, 35.$$

Hence by Theorem 4.2.2, we have 10 non-trivial codes. Their minimum distances can be computed using MAGMA or from their generator matrices. This completes the proof.  $\square$

By Theorem 3.3.2 the non-trivial codes  $C_{36,2}$  can correct up to 17 errors,  $C_{36,4}$  can correct maximum of 6 errors,  $C_{36,5}$  and  $C_{36,6}$  both correct 3 errors each,  $C_{36,7}$  and  $C_{36,8}$  correct up to 2 errors each, and finally  $C_{36,9}$  and  $C_{36,10}$  correct up to only one error each.

**Proposition 5.2.8** *Codes  $C_{36,9}$  and  $C_{36,4}$  are even.*

**Proof:** The direct sum of  $C_{36,9}$  and  $C_{36,4}$  form  $C_{36,11}$  which is the dual code of a repetition code now it follows from Corollary 4.2.3 that the codes are even.  $\square$

**Proposition 5.2.9** *Let  $G = U(3, 3)$  and  $\mathbb{F} = \mathbb{F}_2$ . Then there are no linear codes of length 36 over  $\mathbb{F}$  accepting  $G$  as primitive permutation group that are self dual.*

**Proof:** Let  $C$  be the self dual code of dimension  $k$  then using Remark 3.2.9,  $36 - k = k$  which implies  $k = 18$  but there is no submodule of  $P_1(2)$  that has a dimension 18 meaning such a code does not exist. Hence the result follows.  $\square$

Results of Proposition 5.2.9 can be generalised by Theorem 5.2.10.

**Theorem 5.2.10** *Let  $C$  be a binary code of length  $m$  and dimension  $k$ . Then  $C$  is self dual if and only if  $k = \frac{m}{2}$ .*

**Proof:** Let  $C = [m, k, d]$  be a code and  $C^\perp = [m, m - k, d']$  be it's dual code since  $C$  is self dual we must have  $m - k = k$  resulting in  $k = \frac{m}{2}$ .

Conversely: Let  $C = [m, k, d]$  be a code and  $C^\perp = [m, m - k, d']$  be it's dual code suppose  $k = \frac{m}{2}$ , then

$$m - k = m - \frac{m}{2} = \frac{m}{2}. \quad \square$$

Table 5.6: Weight Distribution of some of the codes of Length 36

Naming	Code	Weight distribution
$C_{36,3}$	[36, 7, 16]	$0^1, 16^{63}, 20^{63}, 36^1$
$C_{36,4}$	[36, 8, 14]	$0^1, 14^{36}, 16^{63}, 18^{56}, 20^{63}, 22^{36}, 36^1$
$C_{36,5}$	[36, 14, 8]	$0^1, 8^{63}, 12^{441}, 14^{2304}, 16^{3591}, 18^{3584}, 20^{3591}, 22^{2304},$ $24^{441}, 28^{63}, 36^1$

We chose few codes of length 36 that we will work on, and give their weight distributions on Table 5.6.

**Proposition 5.2.11** *The code  $C_{36,3}$  is doubly even.*

**Proof:** Using MAGMA we find that the weight distribution of  $C_{36,3}$  is

$$\{0^1, 16^{63}, 20^{63}, 36^1\}$$

and we note all weights of above mentioned codes are divisible by 4. □

**Proposition 5.2.12** *Let  $G$  be a unitary group  $U(3, 3)$  and  $M$  a permutation module of dimension 36 invariant under  $G$ . Then all non-trivial codes of length 36 are even.*

**Proof:**

- i We proved in Proposition 5.2.11 that  $C_{36,3}$  is doubly even which implies that it is even.
- ii  $C_{36,9}$  and  $C_{36,4}$  are even as proved on Proposition 5.2.8.
- iii Using MAGMA we note that the weight distributions of all the codes are divisible by 2 hence it follows that they are all even.

Also using the lattice diagram we can see that all the codes are contained in the dual of the repetition code so by Corollary 4.2.3 all the codes are even. □

**Proposition 5.2.13** *Let  $G$  be a primitive group of degree 36 of the unitary group  $U(3, 3)$  and  $C_{36,1}, C_{36,2}, C_{36,3}, C_{36,4}, C_{36,5}, C_{36,6}, C_{36,7}, C_{36,8}, C_{36,9}, C_{36,10}, C_{36,11}$  and  $C_{36,12}$  of dimension 0, 1, 7, 8, 14, 15, 21, 22, 28, 29, 35 and 36 respectively, then the following holds.*

- i.  $C_{36,1}$  is a trivial code.*
- ii.  $C_{36,2}$  is a repetition code and its dual is  $C_{36,11}$ .*
- iii. There is no non-trivial code of length 36 that can correct more than 7 errors.*

**Proof:** We know that a trivial code is of the only code with the structure  $[n, 0, n]$ , repetition code is of the form  $[n, 1, n]$  and its dual is of form  $[n, n - 1, 2]$ . By Theorem 3.3.2 if  $C$  is code with length  $d$  then  $C$  can correct up to  $\frac{d-1}{2}$  errors and the code of length 36 with the largest minimum distance has  $d = 16$  so it can correct only up to 7 errors.  $\square$

Code  $C_{36,3}$  is more optimal because it has larger minimum distance compared to others so it can detect and correct more errors, making it more optimal for error-prone channels. The error correcting capability depends on the size of the minimum distance, but we must also consider the ratio which is the ratio of the number of information bits to the total number of bits in a codeword denoted by  $R = \frac{k}{n}$ . Higher rates mean more efficient use of the available channel bandwidth. In general the optimal code that achieves the best trade off in between error correction and error detection, computational complexity and bandwidth efficient (higher rate). So finding an optimal codes is not an easy task it depends on the goals and constraints of communication system, because optimality needs, larger minimum distance (for good error correcting capability), bigger ratio, which requires bigger length  $k$  and smaller dimension  $m$  but at the same time larger dimension is required for larger bits of codeword.

Table 5.7 shows some of the properties of the non-trivial codes we constructed using MAGMA the last column shows whether the automorphism is primitive or not.

Table 5.7: non-trivial codes of length 36

Code	Parameters	$\text{Aut}(C_i)$	Primitivity of $C_i$
$C_{36,3}$	[36, 7, 16]	$S(2, 6)$	yes
$C_{36,4}$	[36, 8, 14]	$G(2, 2)$	yes
$C_{36,5}$	[36, 14, 8]	$G(2, 2)$	yes
$C_{36,6}$	[36, 15, 8]	$S(2, 6)$	yes
$C_{36,7}$	[36, 21, 6]	$S(2, 6)$	yes
$C_{36,8}$	[36, 22, 6]	$G(2, 2)$	yes
$C_{36,9}$	[36, 28, 4]	$G(2, 2)$	yes
$C_{36,9}$	[36, 29, 4]	$S(2, 6)$	yes

### Designs held by support of codewords in $C_{36,i}$

Suppose that  $w_n$  is a codeword of nonzero weight  $m$  in a non-trivial code  $C = C_{36,i}$  where  $3 \leq i \leq 10$ . In this Section we determine the structure of  $(\text{Aut}(C))_{w_m}$ , that is the stabilizer of  $w_m$  in  $\text{Aut}(C)$  where  $w_m = \{c \in C : w(c) = m\}$ .

The procedure is the same as that of degree 28 as a results of more codes and larger weight distribution we have many supports resulting in large number of support designs. In this section we look at some of the support designs and stabilizers.

**Proposition 5.2.14** *Let  $w$  be a codeword of the code  $C_{36,i}$   $3 \leq i \leq 10$  of weight  $m$  and  $A = \text{Aut}(C_i)$  if  $w_m \neq \emptyset$  then the action of  $A$  on  $W_m(C_1)$  is transitive. The stabilizer of  $w = W_m(C_i)$  in  $A$  is a maximal subgroup of  $U(3, 3)$  and the support designs constructed from these codes are shown in Table 5.8 and Table 5.9.*

Table 5.8: Stabilizer and support designs from [36, 8, 14]

$m$	$s :=  w_m $	stabilizer	Maximal in $A$	Design
14	36	$L(3, 2) : 2$	yes	$1 - (36, 14, 14)$

16	63	$M_8.S_4$	no	$1 - (36, 16, 28)$
18	56	$3_+^{1+2} : 8$	yes	$1 - (36, 18, 28)$
20	63	$M_8.S_4$	no	$1 - (36, 20, 35)$
22	36	$L(3, 2) : 2$	no	$1 - (36, 22, 22)$

Table 5.9: Some stabilizers and support designs from [36, 14, 8]

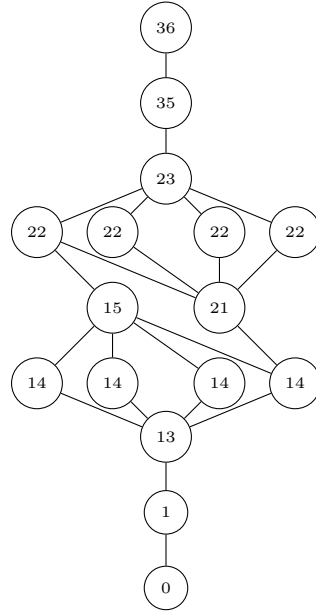
<b>m</b>	$s :=  w_m $	<b>stabilizer</b>	<b>Maximal in <math>A</math></b>	<b>Design</b>
8	63	$M_8.S_4$	yes	$1 - (36, 8, 14)$
12	441	$M_8.S_4$	yes	$1 - (36, 12, 147)$
14	2304	$L(3, 2) : 2$	no	$1 - (36, 14, 896)$
16	3591	$M_8.S_4$	yes	$1 - (36, 16, 1596)$
18	3584	$3_+^{1+2} : 8$	no	$1 - (36, 18, 1792)$
28	63	$M_8.S_4$	yes	$1 - (36, 28, 49)$

**Proof:** Values of  $m$  and  $s$  are given in the weight distribution of the codes. Using Theorem 3.4.6 the designs from the supports of these codes are of the form  $1 - (n, m, \lambda)$  where  $n$  is the length of the code and  $\lambda = ms/n$  and the result follows.  $\square$

### Non-binary codes from representation of degree 36

As discussed in Section 5.1  $M_3 \cong L(2, 7)$  is perfect therefore we can find linear codes of length 36 in  $GF(q)$  where  $q$  the prime factor of  $|G|$ . The method presented here can be used to find non-binary codes. In this section we find codes of length 36 in  $GF(3)$  and  $GF(7)$ . Codes in  $GF(3)$  are called ternary codes. In the field  $\mathbb{F}_3$  the permutation module of degree 36 breaks into submodules of dimensions 0, 1,  $14^4$ , 15, 21,  $22^4$ , 23, 35 and 36. The submodule lattice is shown in Figure 5.3, which also shows that the only irreducible code is of dimension 1. According to Theorem 4.2.2 it suffices to find codes with those dimensions. Due to triviality we exclude the code of dimension 0 and 36, therefore we have 11 non-trivial

Figure 5.3: Submodule lattice for a 36-dimensional representation in  $\mathbb{F}_3$



ternary codes accepting  $U(3, 3)$  as primitive permutation group. We have the length and the dimensions of the ternary codes we left with determining minimum distances to have the complete codes. Using MAGMA we find that the non-trivial codes are as follows.

- |                                  |                                    |
|----------------------------------|------------------------------------|
| i $[36, 13, 12]_3$               | iv $[36, 15, 8]_3$                 |
| ii 3 of the form $[36, 14, 8]_3$ | v $[36, 21, 6]_3$                  |
| iii $[36, 14, 12]_3$             | vi 4 of the form $[36, 22, 6]_3$ . |

Looking at the weight distribution computed by MAGMA we notice that 2 of the  $[36, 14, 8]_3$  codes are isomorphic, their dual  $[36, 22, 4]_3$  are not isomorphic, this means that if the codes are isomorphic it does not imply that their dual codes are also isomorphic.

We also notice that all the non-trivial codes have the minimum distance  $d \geq 3$  therefore from definition of projective codes, all the non-trivial ternary codes under  $U(3, 3)$  are projective.

Table 5.10 shows weight distribution of some of the ternary code.

Table 5.10: Weight Distribution of some of the codes of Length 36

Code	Weight distribution
$[36, 13, 12]_3$	$0^1, 12^{882}, 15^{3024}, 18^{77196}, 21^{381528}, 24^{648270}, 27^{421568},$ $30^{58716}, 33^{3024}, 36^{114}$
$[36, 14, 12]_3$	$0^1, 12^{2520}, 15^{8640}, 18^{237720}, 21^{1125504}, 24^{1973160},$ $27^{1242080}, 30^{185472}, 33^{7560}, 36^{3132}$

**Theorem 5.2.15** *Let  $C$  be a code over  $\mathbb{F}_3^n$ . Then every code word  $c$  has a weight divisible by 3 if and only if  $C$  is self orthogonal.*

**Proof:** See [2, Theorem 1.4.8]. □

**Proposition 5.2.16** *The only self-orthogonal ternary codes accepting  $U(3, 3)$  as the primitive permutation group are  $[36, 13, 12]_3$  and  $[36, 14, 12]_3$ .*

**Proof:** The weight distributions of  $[36, 13, 12]_3$  and  $[36, 14, 12]_3$  are,

$$\{0^1, 12^{882}, 15^{3024}, 18^{77196}, 21^{381528}, 24^{648270}, 27^{421568}, 30^{58716}, 33^{3024}, 36^{114}\}$$

and

$$\{0^1, 12^{2520}, 15^{8640}, 18^{237720}, 21^{1125504}, 24^{1973160}, 27^{1242080}, 30^{185472}, 33^{7560}, 36^{3132}\}$$

respectively, the weight distribution shows that all the code words have weight divisible by 3, so by Theorem 5.2.15 these codes are self-orthogonal and they are the only ternary codes with all the code words having weight divisible by 3, therefore they are the only self-dual ternary codes. □

**Proposition 5.2.17** *The ternary repetition code accepting  $G = U(3, 3)$  as the primitive permutation group is  $[36, 1, 36]_3$  and its dual is  $[36, 35, 2]_3$ .*

**Proof:** We know that the only perfect maximal subgroup of  $U(3, 3)$  is isomorphic to  $L(2, 7)$  and is of degree 36 therefore we have one repetition code of ternary type with  $n = 36$ . By Lemma 3.2.8, we have the code  $[36, 1, 36]_3$  and its dual code is of the form



$$[n, n - 1, 2]_q = [36, 35, 2]_3. \quad \square$$

In the field  $\mathbb{F}_7$  the 36 dimensional representation splits into 5 non-trivial submodules of length 14, 15, 21, 22 and 35. So we have 5 non-trivial codes  $[36, 14, 8]$ ,  $[36, 15, 8]$ ,  $[36, 21, 6]$ ,  $[36, 22, 6]$  and  $[36, 35, 2]$  which are all even and all projective. Table 5.11 shows the  $Aut(C)$  of all the non-trivial ternary codes.

Table 5.11: Automorphism and permutation automorphism groups of non-trivial ternary codes.

<b>Parameters</b>	$Aut(C)$	<b>Primitivity of <math>Aut(C)</math></b>
$[36, 13, 8]_3$	$2 \times U(3, 3) : 2$	yes
$[36, 14, 8]_3$	$W(E7)$	yes
$[36, 14, 12]_3$	$U(3, 3) : 2$	yes
$[36, 15, 8]_3$	$W(E7)$	yes
$[36, 21, 6]_3$	$W(E7)$	yes
$[36, 22, 6]_3$	$U(3, 3) : 2$	yes
$[36, 23, 6]_3$	$2 \times U(3, 3) : 2$	yes

### 5.2.3 Representations of degree 63

We have 2 representations of degree 63 under  $U(3, 3)$  we shall name them 63a and 63b to distinguish them and later compare their codes. We construct 63-dimensional permutation module invariant under  $U(3, 3)$  acting on a set of degree 36 by letting the permutation module be our working module and find all submodules.

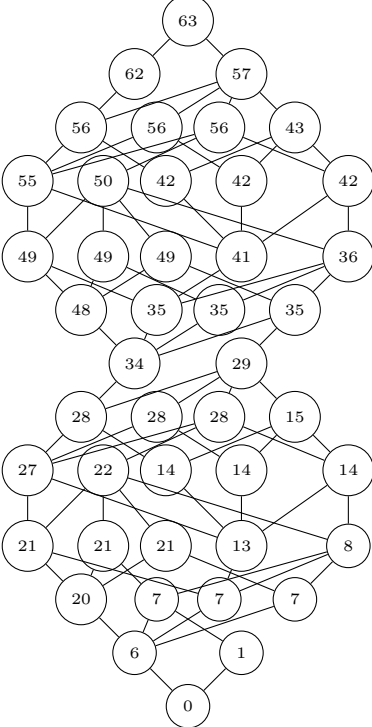
#### Representation 63a

Looking at the first permutation module of degree 63, and letting it be our working module and find all its submodules. The module splits into 44 submodules of dimension

$$0, 1, 6, 7^3, 8, 13, 14^3, 15, 20, 21^3, 22, 27, 28^3, 29, 34, 35^3, 36, 41, 42^3, 43, 48, 49^3, 50, 55, 56^3, 57, 62, 63.$$

The submodule lattice of this representation is shown in Figure 5.4, showing that codes of dimension 6 and 1 are irreducible. According to Theorem 4.2.2 there are 44 codes of length

Figure 5.4: Submodule lattice for a 63-dimensional representation



63 and dimensions given above. So it is left to find the minimum distance of those linear codes so that the parameters will be complete.

**Proposition 5.2.18** *Let  $G = U(3, 3)$  and  $H$  a representation of degree 63 under  $G$  then there exist codes  $C_{63a,0} = [63, 0, 63]$ ,  $C_{63a,1} = [63, 1, 63]$ ,  $C_{63a,42} = [63, 62, 2]$  and  $C_{63a,43} = [63, 63, 1]$ . Moreover  $C_{63a,42}$  is the dual code of  $C_{63a,1}$ .*

**Proof:** By definition for every permutation module of dimension  $n$  there exist 2 trivial codes  $[n, 0, n]$  and  $[n, n, 1]$  hence for  $H$  we have  $[63, 0, 63]$  and  $[63, 63, 1]$ . Then by Lemma 3.2.8 we note that there is a code  $[n, 1, n]$  and its dual is  $[n, n - 1, 2]$  hence we have  $[63, 1, 63]$  and its dual is  $[63, 62, 2]$  and the result follows. □

Codes under this representation are of the form  $[63, n, d]$  where  $n$  is the dimension of the submodule and  $d$  is the minimum weight. Since we have dimensions of all the codes, we are left with finding the minimum weight.

Using MAGMA to compute the minimum weight we find that some of the non-trivial codes are as follows.

- |   |   |
|---|---|
| 1. $C_{63a,2} = [63, 6, 32]$                                | 16. $C_{63a,17} = [63, 27, 12]$                   |
| 2. $C_{63a,3} = [63, 7, 28]$                                | 17. $C_{63a,18} = [63, 28, 12]$                   |
| 3. $C_{63a,4} = [63, 7, 27]$                                | 18. $C_{63a,19} = [63, 28, 12]$                   |
| 4. $C_{63a,5} = C_{63a,2} \oplus C_{63a,1} = [63, 7, 31]$   | 19. $C_{63a,20} = [63, 28, 11]$                   |
| 5. $C_{63a,6} = C_{63a,3} \oplus C_{63a,1} = [63, 8, 27]$   | 20. $C_{63a,21} = [63, 29, 11]$                   |
| 6. $C_{63a,7} = [63, 13, 20]$                               | 21. $C_{63a,22} = C_{63a,21}^\perp = [63, 34, 8]$ |
| 7. $C_{63a,8} = [63, 14, 20]$                               | 22. $C_{63a,23} = C_{63a,20}^\perp = [63, 35, 7]$ |
| 8. $C_{63a,9} = [63, 14, 14]$                               | 23. $C_{63a,24} = C_{63a,19}^\perp = [63, 35, 8]$ |
| 9. $C_{63a,10} = C_{63a,7} \oplus C_{63a,1} = [63, 14, 20]$ | 24. $C_{63a,25} = C_{63a,18}^\perp = [63, 35, 7]$ |
| 10. $C_{63a,11} = [63, 15, 14]$                             | 25. $C_{63a,26} = C_{63a,17}^\perp = [63, 36, 7]$ |
| 11. $C_{63a,12} = [63, 20, 16]$                             | 26. $C_{63a,27} = C_{63a,16}^\perp = [63, 41, 8]$ |
| 12. $C_{63a,13} = [63, 21, 16]$                             | 27. $C_{63a,28} = C_{63a,15}^\perp = [63, 42, 8]$ |
| 13. $C_{63a,14} = [63, 21, 15]$                             | 28. $C_{63a,29} = C_{63a,14}^\perp = [63, 42, 7]$ |
| 14. $C_{63a,15} = [63, 21, 15]$                             | 29. $C_{63a,30} = C_{63a,13}^\perp = [63, 42, 7]$ |
| 15. $C_{63a,16} = [63, 22, 15]$                             | 30. $C_{63a,31} = C_{63a,12}^\perp = [63, 43, 7]$ |

31.  $C_{63a,32} = C_{63a,11}^\perp = [63, 48, 4]$                       36.  $C_{63a,37} = C_{63a,6}^\perp = [63, 55, 4]$
32.  $C_{63a,33} = C_{63a,10}^\perp = [63, 49, 4]$                       37.  $C_{63a,39} = C_{63a,5}^\perp = [63, 56, 3]$
33.  $C_{63a,34} = C_{63a,9}^\perp = [63, 49, 4]$                       38.  $C_{63a,40} = C_{63a,4}^\perp = [63, 56, 4]$
34.  $C_{63a,35} = C_{63a,8}^\perp = [63, 49, 3]$                       39.  $C_{63a,41} = C_{63a,3}^\perp = [63, 56, 3]$
35.  $C_{63a,36} = C_{63a,7}^\perp = [63, 50, 3]$                       40.  $C_{63a,42} = C_{63a,2}^\perp = [63, 57, 3]$

Table 5.12: Weight Distribution of some of the codes of Length 63

Naming	Code	Weight distribution
$C_{63a,2}$	$[63, 6, 32]$	$0^1, 32^{63}$
$C_{63a,3}$	$[63, 7, 28]$	$0^1, 28^{36}, 32^{63}, 36^{28}$
$C_{63a,7}$	$[63, 13, 20]$	$0^1, 20^{252}, 24^{378}, 28^{1800}, 32^{3591}, 36^{2044}, 40^{126}$

Having that from MAGMA the weight distribution of  $C_{63a,2}$  is  $\{0^1, 32^{63}\}$  and  $C_{63a,5} = C_{63a,2} \oplus C_{63a,1}$  where  $C_{63a,1}$  is an all one code we can find the weight distribution of  $C_{63a,5}$  by adjoining the ones-vectors of  $C_{63a,2}$  to get  $\{0^1, 31^{63}, 32^{63}, 63^1\}$  and we can see that the minimum distance is 31 and the code is not even. In a similar way we can find weight distributions of  $C_{63a,6} = C_{63a,3} \oplus C_{63a,1} = [63, 8, 27]$  and  $C_{63a,10} = C_{63a,7} \oplus C_{63a,1} = [63, 14, 20]$  and all other codes that can be written as the direct sum of another code and the repetition code.

**Proposition 5.2.19** *Let  $G = U(3, 3)$  and  $H$  be a permutation module of degree 63. Then only the following codes are not even.*

1.  $C_{63a,42}$     4.  $C_{63a,36}$
2.  $C_{63a,41}$     5.  $C_{63a,35}$
3.  $C_{63a,39}$     6.  $C_{63a,34}$

- |                  |                  |
|------------------|------------------|
| 7. $C_{63a,31}$  | 16. $C_{63a,16}$ |
| 8. $C_{63a,30}$  | 17. $C_{63a,15}$ |
| 9. $C_{63,29}$   | 18. $C_{63a,14}$ |
| 10. $C_{63a,26}$ | 19. $C_{63a,11}$ |
| 11. $C_{63a,25}$ | 20. $C_{63a,10}$ |
| 12. $C_{63a,23}$ | 21. $C_{63a,8}$  |
| 13. $C_{63a,21}$ | 22. $C_{63a,6}$  |
| 14. $C_{63a,20}$ | 23. $C_{63a,5}$  |
| 15. $C_{63a,18}$ | 24. $C_{63a,4}$  |

**Proof:** According to Corollary 4.2.3 the only even codes are those contained in the dual of the repetition code, which is a code of dimension 62. Using the lattice structure we note that the codes mentioned are not contained in the code of length 62, hence they are not even and all others are contained in the code of length 62 there they are even.  $\square$

Based on codes  $C_{63a,8}$  and  $C_{63a,10}$  we notice that codes may have same parameters but have different structures and hence not isomorphic, since they have different weight distributions.

**Proposition 5.2.20** *The following codes are doubly even.*

- i*  $C_{63a,2}$
- ii*  $C_{63a,3}$
- iii*  $C_{63a,7}$

**Proof:** Using MAGMA to compute the weight distributions we notice that the weight distributions of the above codes are  $\{0^1, 32^{63}\}$ ,  $\{0^1, 28^{36}, 32^{63}, 36^{28}\}$  and  $\{0^1, 20^{252}, 24^{378}, 28^{1800}, 32^{3591}, 36^{2044}, 40^{126}\}$  respectively and we note that all weights are divisible by four, therefore the codes are doubly even.  $\square$

Codes mentioned in Proposition 5.2.20 are not the the only doubly even codes, we just mentioned few for simplicity.

With the application of Theorem 3.3.2 we note that the code  $C_{63a,2}$  corrects more errors ( $\lfloor \frac{32-1}{2} \rfloor = 15$ ) than any other 63 dimensional codes.

**Proposition 5.2.21** *every non-trivial codes under 63a is projective.*

**Proof:** We note that all the non trivial codes have a minimum distance  $d \geq 3$ , therefore the dual distances of all the codes is at least three hence all the codes are projective.  $\square$

**Proposition 5.2.22** *Codes  $C_{63a,2}$  and  $C_{63a,7}$  are self orthogonal.*

**Proof:** We note the dual code of  $C_{63a,2}$  and  $C_{63a,7}$  are  $C_{63a,42}$  and  $C_{63a,36}$  respectively, using the lattice diagram we note that those codes are subsets of their dual codes, therefore the codes are self orthogonal.  $\square$

Table 5.13 shows properties of some non-trivial codes of length 63 we constructed. We choose codes with different automorphism groups, all other codes has either of those group as their automorphism groups. The computations were based on MAGMA.

Table 5.13: non-trivial codes of length 63

Code	parameters	$\text{Aut}(C_i)$	Primitivity of $C_i$
$C_{63a,2}$	[63, 6, 32]	$L(6, 2)$	yes
$C_{63a,3}$	[63, 7, 28]	$S(2, 6)$	yes
$C_{63a,7}$	[63, 13, 20]	$U(3, 3)$	yes

### Designs held by supports of codewords in $C_{63a,i}$

Suppose that  $w_n$  is a codeword of nonzero weight  $m$  in a non-trivial code  $C = C_{63a,i}$  where  $3 \leq i \leq 42$ . In this section we determine the structure of  $(\text{Aut}(C))_{w_m}$ , that is the stabilizer of  $w_m$  in  $\text{Aut}(C)$  where  $w_m = \{c \in C : w(c) = m\}$ .

The procedure is the same as that of degree 28 as a results of more codes and larger weight

distribution we have too many supports resulting in large number of support designs. In this section we look on some of the support designs and stabilizers.

**Proposition 5.2.23** *Let  $w$  be a codeword of the code  $C_{36,i}$ ,  $3 \leq i \leq 42$  of weight  $m$  and  $A = \text{Aut}(C_i)$ . If  $w_m \neq \emptyset$  then the action of  $A$  on  $W_m(C_1)$  is transitive. The stabilizer of  $w = W_m(C_i)$  in  $A$  is a maximal subgroup of  $U(3,3)$  and the support designs constructed from these codes are shown in Table 5.14 and Table 5.15.*

Table 5.14: Stabilizers and support designs from [63, 6, 32]

<b>m</b>	$s :=  w_m $	<b>stabilizer</b>	<b>maximal in <math>A</math></b>	<b>Design</b>
32	63	$2^5 : L(5, 2)$	yes	$1 - (63, 32, 32)$

Table 5.15: Stabilizers and support designs from [63, 7, 28]

<b>m</b>	$s :=  w_m $	<b>stabilizer</b>	<b>maximal in <math>A</math></b>	<b>Design</b>
28	36	$S_8$	yes	$1 - (63, 28, 16)$
32	63	$2^2 : S_6$	yes	$1 - (63, 32, 32)$
36	28	$U(4, 2) : 2$	yes	$1 - (63, 36, 16)$

**Proof:** Values of  $m$  and  $s$  are given in the weight distribution of the codes. Using Theorem 3.4.6 the designs from the supports of these codes are of the form  $1 - (n, m, \lambda)$  where  $n$  is the length of the code and  $\lambda = ms/n$ . To find stabilizers we used MAGMA and [43], they show all the maximal subgroups of  $A = S(2, 6)$  and the results follows.  $\square$

### Representation 63b

We now look at the second permutation module of degree 63, and letting it be our working module and find all its submodules. The module splits into 28 submodules of dimension

$$0, 1, 14, 15, 20, 21^3, 22, 27, 28^3, 29, 34, 35^3, 41, 42^3, 43, 48, 49, 62, 63.$$

According to Theorem 4.2.2 there are 28 codes of length 63 and dimensions given above. So it is left to find the minimum distance of those linear codes so that the parameters will be complete. Trivial codes under  $63b$  are similar to those under  $63a$  mentioned in Proposition 5.2.18. The submodule lattice of this representation as shown in Figure 5.5, in which we note that the irreducible submodules are of dimension 14 and 1, resulting in irreducible codes of those dimensions.

Figure 5.5: Submodule lattice for a 63-dimensional representation

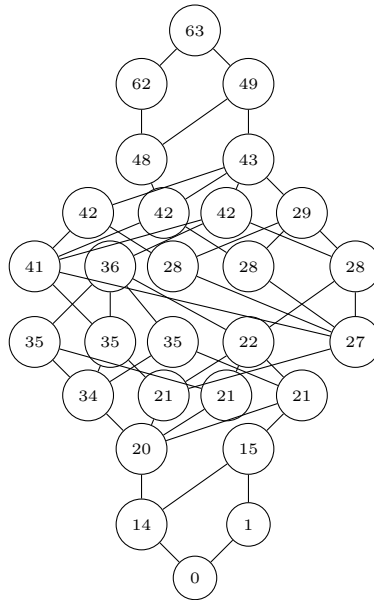


Table 5.16 show some codes of length 63 and their weight distribution.

Table 5.16: Weight Distribution of some of the codes of Length 63

Code	Weight distribution
$[63, 14, 16]$	$0^1, 16^{126}, 24^{1596}, 28^{2880}, 32^{7497}, 36^{252}$
$[63, 20, 16]$	$0^1, 16^{693}, 20^{3024}, 24^{78456}, 28^{278064}, 32^{420651}, 36^{222768}, 40^{41832}, 48^{63}$
$[63, 15, 16]$	$0^1, 16^{126}, 23^{252}, 24^{1596}, 27^{4032}, 28^{2880}, 31^{7497}, 32^{7497}, 35^{2880}, 36^{4032}, 39^{1593}, 40^{252}, 63^1$

Corollary 4.2.3 says that the codes are even if and only if they are contained in the dual of



repetition code, so using the lattice diagram we can tell which codes are even and we use MAGMA to find their minimum weight and they are

- |                 |                  |
|-----------------|------------------|
| 1. [63, 48, 4]  | 6. [63, 27, 12]  |
| 2. [63, 42, 6]  | 7. [63, 34, 8]   |
| 3. [63, 41, 6]  | 8. [63, 21, 16]  |
| 4. [63, 28, 12] | 9. [63, 20, 16]  |
| 5. [63, 35, 8]  | 10. [63, 14, 16] |

For optimality we have 3 codes with minimum distance 16 (i.e [63, 21, 16], [63, 20, 16],[63, 14, 16]) but the code [63, 21, 16] is more optimal than those 2 because it has larger dimension, resulting in larger ratio.

**Proposition 5.2.24** *The following codes are doubly even.*

1. [63, 14, 16]
2. [63, 20, 16]

**Proof:** The codes, have the weight distribution  $\{0^1, 16^{126}, 24^{1596}, 28^{2880}, 32^{7497}, 36^{252}\}$  and  $\{0^1, 16^{693}, 20^{3024}, 24^{78456}, 28^{278064}, 32^{420651}, 36^{222768}, 40^{41832}, 44^{3024}, 48^{63}\}$ , respectively and we note that all the weights are divisible by 4 therefore the codes are doubly even.  $\square$

Now having the code [63, 14, 16] and its weight distribution, and from the lattice diagram we note we have code  $[63, 15, d] = [63, 14, 16] + \langle J \rangle$ , where  $\langle J \rangle$  is the all one code. The weight distribution of  $[63, 15, d]$  can be determined by adjoining the ones-vectors to get

$$\{0^1, 16^{126}, 23^{252}, 24^{1596}, 27^{4032}, 28^{2880}, 31^{7497}, 32^{7497}, 35^{2880}, 36^{4032}, 39^{1593}, 40^{252}, 47^{126}, 63^1\}$$

and hence we can get the minimum distance (16) from the weight distribution and we have the code [63, 15, 16] which is not even. Which is contained in its dual code  $C^\perp = [63, 44, 3]$  therefore the code [63, 44, 3] is self-orthogonal, also both codes have  $d \geq 3$  therefore they

are both projective. Table 5.17 shows properties of some non-trivial codes of length 63 we constructed. We choose codes with different automorphism groups, all other codes has either of those group as their automorphism groups. The computations were based on MAGMA the last column tell whether  $Aut(C_i)$  is primitive or not.

Table 5.17: Non-trivial codes of length 63

parameters	$Aut(C_i)$	Primitivity of $Aut(C_i)$
[63, 14, 16]	$U(3, 3) : 2$	yes
[63, 15, 16]	$U(3, 3) : 2$	yes
[63, 20, 16]	$U(3, 3) : 2$	yes

We notice that all non-trivial codes under representation 63b have  $U(3, 3) : 2$  as their automorphism group.

### Designs held by support of codewords in $C_{63b,i}$

Table 5.18 shows designs held by support of the code [63, 14, 16] with weight distribution  $\{0^1, 16^{126}, 24^{1596}, 28^{2880}, 32^{7497}, 36^{252}\}$ .

Table 5.18: Stabilizers and support designs from [63, 14, 16]

<b>m</b>	$s :=  w_m $	stabilizer	maximal in $A$	Design
16	126	$S_8$	no	$1 - (63, 16, 32)$
24	1596	$2^2 : S_6$	no	$1 - (63, 24, 608)$
28	2880	$U(4, 2) : 2$	no	$1 - (63, 28, 1280)$
32	7497	$S_3$	no	$1 - (63, 32, 2304)$
36	252	$M_8 : S_4$	no	$1 - (63, 36, 160)$

# Conclusion

We found binary linear codes of length 28, 36 and 63 and there are no self dual codes accepting  $U(3, 3)$  as primitive permutation group but there is a number of self orthogonal and projective codes. There is also set of Golay and ternary codes all of length 36 invariant under  $U(3, 3)$ . Code  $C$  and its dual  $C^\perp$  always have the same automorphism group and the automorphism group always contains  $U(3, 3)$ . The order of our group  $U(3, 3)$  divides the order of all the automorphism groups of codes invariant under  $U(3, 3)$ . Each code has weight distribution and supports which we used to find designs corresponding to each supports and one code can support more than one design.

# Bibliography

- [1] E. Abbe, *Coding theory and coding techniques*, Princeton Education, pp01, (2020).
- [2] M.M. Al-Ashker, *Coding theory lectures*, University of Gaza, Palestine.
- [3] J.L. Alperin, and R.B Bell, *Groups and representations*, Springer, (1995).
- [4] E.F. Assmus, and J.D Key, *Affine and projective planes*, Discrete. Math., vol. 83, pp161-187, (1990).
- [5] E.F. Assmus, and J.D Key, *Designs and their codes*, volume 103 of Cambridge Tracts in Mathematics, pp389-419, (1993).
- [6] A.B.M. Basheer, *Representation theory of finite groups*, AIMS, (2006).
- [7] N.L. Biggs, and A.T. White, *Permutation groups and combinatorial structure*, Series 33 lecture notes of Cambridge University Press, (1979).
- [8] S. Biswas, *Introduction to coding theory: Basic codes and Shannon's theorem*. pp6, (2011).
- [9] L. Bolcar, *On the weights of linear codes and their dual*, Digital commons, (2020).
- [10] W. Bosma, J. Cannon, and C. Playoust, *The MAGMA algebra system I: The user language*, journal of symbolic computation, 24(3-4), pp235-265, (1997).
- [11] P. L. H. Brooke, *On the Steiner system  $S(2, 4, 28)$  and codes associated with the simple group of order 6048*, J. Algebra 97, (1985).

- [12] R. Brauer, *On modular characters of groups*, Annals of mathematics 42, pp556-590,(1941)
- [13] S.K. Buddha, *Hamming and Golay Codes*, Indiana University Terre Haute, pp04, (2011).
- [14] L. Chikamai, J. Moori, and B. G. Rodrigues, *Linear codes obtained from 2-modular representations of some finite simple groups*. Ph.D. thesis, University of Kwazulu Natal, (2012).
- [15] J.H. Conway, R.T. Curtis, S.P. Norton, R.A. Parker, and R.A. Wilson, *Atlas of Finite Groups*, (1985).
- [16] D. Crnković, V. Mikulić, *On some combinatorial structures constructed from the groups  $L(3, 5)$ ,  $U(5, 2)$  and  $S(6, 2)$* , Int. J Comb. 2011 (2011). <http://dx.doi.org/10.1155/2011/137356>. Article ID 137356, 12 pages
- [17] D. Crnković, and V. Mikulić, *Unitals, projective planes and other combinatorial structures constructed from the unitary groups  $U(3, q)$ ,  $q = 3, 4, 5, 7$* , Ars Combin. 110 pp3–13, (2013).
- [18] R.C. Daileda, *Modules and vector space*, (2017).
- [19] M.R. Darafsheh, B. G. Rodrigues, and A. Saeidi, *On linear codes constructed from finite groups with a trivial Schur multiplier*, Mathematical communication, 28(2023) 85-104, (2013).
- [20] C. Ding and C. Tang, *Designs from linear codes*, Second edition, World Scientific, (2022).
- [21] D.S. Dummit, R.M. Foote, *Abstract algebra*, John Wiley and sons, Inc, (2004).
- [22] The GAP Group, *GAP - Groups, Algorithms, and Programming*, Version 4.12.2; 2022 (<https://www.gap-system.org>)

- [23] J. Grossman, *Coding theory: introduction to linear codes and applications*. Insight: River Academic Journal, 4(2), pp1-17, (2008).
- [24] A.R. Hurson, *AI and cloud computing*, pp150, (2021).
- [25] G. James, and M. Leibeck, *Representations and characters of groups*, Second edition, Cambridge, (2003).
- [26] J.D. Key, and J. Moori, *Codes, designs and graphs from the janko groups  $J_1$  and  $J_2$* , J. Combin. Math. Combin. Comput. 40 pp143–159, (2002).
- [27] J.D. Key, and J. Moori, *Correction to: Codes, designs and graphs from the Janko groups*, J. Combin. Math. Combin. Comput. 64, (2008).
- [28] W. Knapp, and P. Schmid, *Codes with prescribed permutation automorphism*, J Algebra 67 no.2, pp41-435, (1980).
- [29] R. Lal, *Algebra 2, linear algebra, Galois theory, representation theory, group extension and Schur Multiplier*, Springer, (2017).
- [30] D.C. Lay, S.R. Lay, and J.J. McDonald, *Linear algebra and its application*, Pearson, (2015).
- [31] Y. Lindell, *Introduction to coding theory lecture notes*, Department of Computer Science Bar-Ilan University, Israel, (2010).
- [32] M.G Mahmoudi, *A proof of Krull-Schmidt's theorem for modules*, Shariff University of Technology, (2012).
- [33] V.N. Marani, *Some linear codes, graphs and designs form Matheu groups  $M_{24}$  and  $M_{23}$* . Ph.D thesis University of Kwazulu Natal, (2019).
- [34] J.S. Milne, *Group theory*, v4.00, (2021), [www.jmilne.org/math/](http://www.jmilne.org/math/).
- [35] J. Moori, *AIMS course on further group theory/representation theory*, North West University, Mafikeng, (2011).

- [36] J. Moori, *Finite Groups, Designs and Codes*, Northwest University, Mafikeng (2011).
- [37] R. Pinch, *Coding theory: The first 50 years*, plusadmin, (1997).
- [38] R. Prag, *A brief summary of modular representation theory*, Lecture notes.
- [39] B.G. Rodrigues, *Codes of designs and graphs from finite simple groups*, PhD dissertation, University of Kwazulu Natal, (2002).
- [40] G. Sheng, *On the classification of finite simple groups*, (2022).
- [41] P. Webb, *A course in finite group representation theory*, Cambridge university press 161, (2016).
- [42] S.H. Weintraub, *Representation theory of finite group: Algebra and arithmetic*, American Mathematics Society, (2003).
- [43] R.A. Wilson, R.A. Parker, and J.N. Bray, *Atlas of finite group representations*, <http://brauer.maths.qmul.ac.uk/Atlas/cls/S62/>.

# Appendixes

```
load"u33";
Max:=MaximalSubgroups(G);
F:=GF(2);
i:=3;
m:=Max[i]'subgroup;
a1,g,a3:=CosetAction(G,m);
M:=Stabilizer(g,1);
v:=#g/#m;

P:=PermutationModule(g,F);s:=Submodules(P);;

I:=IrreducibleModules(g,F);
c:=[];cc:=[];

for i in [1..#s] do
f:=Morphism(s[i],P);
c[i]:=LinearCode(f);
end for;

for i in [1..#s] do
cc[i]:= [Dimension(P), Dimension(c[i]),
```



```

MinimumDistance(c[i]);
end for;
cc;

t:=[];
sw:=[];

for i in [1..#s] do
sw[i]:=[];
end for;

r:=1;
w:=MinimumDistance(c[r]);
for i in [1..#s] do w:=WeightDistribution(c[i]);
sw[i]:=[w[i][1]:i in [1..#w]];
end for;

A:=[];
for r in [3..#s] do
A[r]:= AutomorphismGroup(c[r]);
end for;sw

r:=1;
C:=c[r];

sw[r];
m:=16;

```

```

    edd:=Words(C,m);
gg:=GSet(A[r],edd);
ob:=Orbits(A[r],gg);#ob;
S:=[];

    // Choose j in [1..#ob] #ob;

    j:=1;
b:=Setseq(ob[j]);
for i in [1..#ob[j]] do
S[i]:=Support(ob[j][i]);
end for;
SS:=Set(S);
st:=Stabilizer(A[r],S);

    D1:=Designj1,v—SS;
D1;

    C=code; m = Weight;
Ac= Aut (C);
Ac:=AutomorphismGroup(C);
edd:=Words(C,m);
gg:=GSet(Ac,edd);
ob:=Orbits(Ac,gg);#ob;
b:=Setseq(ob[1]);
for i in [1..#ob[1]] do
S:=Support(ob[1][i]);
end for;

```

```

SS:=Set(S);
st:=Stabilizer(Ac,S);

j:=j+1; j-1;

D:=Designj2,v—SSi;

for m in sw[r] do
edd:=Words(C,m);
gg:=GSet(A[r],edd);
ob:=Orbits(A[r],gg);#ob;
// #ob is the number of stabilizers
end for;
// fix j in [1..#ob]

b:=Setseq(ob[j]);
S:=Support(b[1]);
st:=Stabilizer(A[r],S);

stb:={};
for x in A[r] do if b[1]^x eq b[1] then
stb:= stb join x;
end if; end for;

stab:= subjA[r]—stbi;

for i in [1..100000] do
v:=Random(C);

```

if Weight(v) le Weight(w) then

w:=v;

end if; end for; Weight(w);