Delft University of Technology
Faculty of Electral Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

ON THE CAP SET PROBLEM
upper bounds on maximal cardinalities of caps in dimensions seven to ten

A thesis submitted to the
Delft Institute of Applied Mathematics
as partial fulfillment of the requirements

for the degree of

BACHELOR OF SCIENCE
in
APPLIED MATHEMATICS

by

Nina D. Versluis

Delft, The Netherlands
July 2017

BSc thesis APPLIED MATHEMATICS

ON THE CAP SET PROBLEM
upper bounds on maximal cardinalities of caps in dimensions seven to ten

Nina D. Versluis

Delft University of Technology

Supervisor

Dr. D. C. Gijswijt

Other committee members

Drs. E. M. van Elderen

Dr. J. A. M. de Groot

July, 2017

Delft

Contents

Preface

This is my bachelor's thesis. It is the result of my bachelor project on the cap set problem. This project is the completion of my bachelor program Applied Mathematics at Delft University of Technology, which I started in 2014.

From start to finish, I was glad with my choice of project. I knew I wanted a project in the field of optimization. To choose between the projects offered within the field was hard, but my love for games led me to the cap set problem, of which the card game SET is a practical (and fun!) example. I was lucky to be assigned to my project of choice, since not all of my fellow students got the same chance.

I want to thank my supervisor Dion for the weekly meetings, the input and the feedback. Furthermore, I want to thank Joost and Emiel for taking place in my assessment committee. Finally, I want to thank my friends and family for their support and interest, although they did not always understand what I was working on. A special thanks goes to Pim, who helped out with the visuals.

Nina Versluis
Delft, July 2017

List of Symbols

d	dimension
$\mathbb{F}_q^d$	d-dimensional affine space over the field $\mathbb{F}_q$
cap	set of points that contains no lines
d-cap	cap in $\mathbb{F}_3^d$
F	field
$PG(d,3)$	d-dimensional projective space over $\mathbb{F}_3$
H	hyperplane
ν_c	$\lvert A \cap H_c \rvert$, with $A \subset \mathbb{F}_3^d$ and H_c a hyperplane
(ν_0, ν_1, ν_2)	hyperplane triple
$(H, \{x_1, ..., x_n\} \subset H \cap A)$	n-marked hyperplane
$x_{\nu_0\nu_1\nu_2}$	variable that represents the number of times a (ν_0, ν_1, ν_2)-triple can occur
t	number of degenerated quartets
$C_d(q)$	maximal cardinality of a cap in $\mathbb{F}_q^d$
$c_d(q)$	ratio $\frac{C_d}{q^d}$
$A(n, \delta, w)$	maximum size of binary code with word length n, minimum distance δ, constant weight w
ζ	primitive third root of unity ($e^{\frac{2}{3}\pi i}$)
U_y	$\sum_{a \in A} \zeta^{a \cdot y}$
$\mathbf{u}$	vector whose entries correspond to $\lvert U_y \rvert, y \neq 0$
S	$\sum_{y \in \mathbb{F}_3^d \setminus \{0\}} \sum_{a_1, a_2, a_3 \in A} \zeta^{(\sum_{i=1}^3 a_i) \cdot y}$
T	$\sum_{y \in \mathbb{F}_3^d} \sum_{a,b,c,d \in A} \zeta^{(a+b-c-d) \cdot y}$

Abstract

This thesis concerns the cap set problem in affine geometry. The problem is illustrated by the card game SET and its geometrical interpretation in ternary affine space. The maximal cardinality of a cap is known for the dimension one to six. For the four lowest dimensions, a maximal cap is constructed and the optimality of its size proven. From there, two recursive methods by Davis and Maclagan [7] and Bierbrauer and Edel [3] are described and applied to obtain upper bounds for the maximal size of caps in dimensions seven to ten. The best found upper bounds are 291, 771, 2070 and 5619, respectively.

1 Introduction

The cap set problem, the main focus of this thesis, is introduced through a practical example: the card game SET. At first, the course of the game is explained. Secondly, the rules of the game are translated into geometrical statements. Finally, the structure of this thesis is given.

1.1 The Card Game SET

In 1974, the card game SET was invented by Marsha Falco, a population geneticist [13]. The SET cards show figures which differ in four characteristics: number, shading, colour and shape. Every characteristic has three possible appearances, which are written down in Table 1. Hence, the number of cards is

$$3 \cdot 3 \cdot 3 \cdot 3 = 3^4 = 81.$$

Characteristic	First Possibility	Second Possibility	Third Possibility
Number	One	Two	Three
Shading	Empty	Spotted	Solid
Colour	Green	Purple	Red
Shape	Rectangle	Oval	Wave

Table 1: Possibilities per characteristic.

In order to play the game, the ground rule must be known.

Rule (SET)**.** Three cards form a SET if each of the four characteristics is identical or totally different.

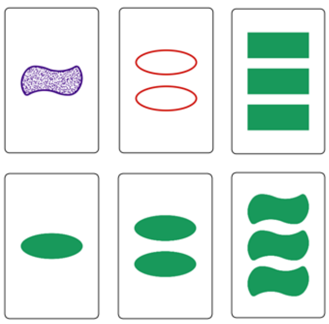

Figure 1: Two triples of SET cards.

The cards in the first triple in Figure 1 differ from each other in each characteristic. Hence, they form a SET. The second triple does not form a SET because the cards are not identical or totally different in the characteristic shape: two of the cards display ovals, the other waves.

A game of SET starts by dealing twelve cards face-up, whereafter players start searching for a SET. The first player to find a SET removes the cards, and three new cards are dealt. The game ends when no SETs can be found while all the cards have been dealt. The winner is the player with the most SETs.

It can occur that there is no SET among the first twelve cards. Then three additional cards are dealt. This can be repeated until a SET can be found. This additional rule raises the following question.

Question. How many cards must be dealt to guarantee the presence of a SET?

In 1971, three years before the invention of the game, Pellegrino already answered this question [12]. Clearly, this was in a different context. That is, in the context of affine geometry.

1.2 Geometrical Interpretation of SET

The characteristics of SET cards can be seen as four different dimensions. Since each characteristic has three possibilities, each dimension has three elements. They correspond as shown in Table 2. Therefore, consider the four-dimensional space over the field of three elements: $\mathbb{F}_3^4$.

Dimension	0 - First Element	1 - Second Element	2 - Third Element
1 - Number	One	Two	Three
2 - Shading	Empty	Spotted	Full
3 - Colour	Green	Purple	Red
4 - Shape	Rectangle	Oval	Wave

Table 2: Translation to geometry.

If $\mathbb{F}_3^4$ describes the entire deck, then every point $y = (y_1, y_2, y_3, y_4) \in \mathbb{F}_3^4$ corresponds to a unique card. For example, the cards in the first triple of Figure 1 correspond to $(0, 1, 1, 2), (1, 0, 2, 1)$ and $(2, 2, 0, 0)$, respectively.

The geometrical interpretation of the SET rule reads as follows. In $\mathbb{F}_3^4$ three points are a SET if they are collinear, which is equivalent to adding up to $(0, 0, 0, 0)$ modulo 3. This definition can be applied on the first triple of Figure 1 of which it is already known that they form a SET. Indeed, $(0, 1, 1, 2) + (1, 0, 2, 1) + (2, 2, 0, 0) = (0, 0, 0, 0)$.

The collinearity of three points can be generalized to apply in any dimension. In general, three points $a, b, c \in \mathbb{F}_3^d$ are collinear if and only if $a + b + c = 0$.

Definition 1 (d-cap)**.** A d-cap is a subset of $\mathbb{F}_3^d$ in which no three points are collinear.

By introducing the term d-cap, the previously raised question can be reformulated and generalized to the following question.

Question. What is the maximal cardinality of a d-cap?

This question is also known as the cap set problem. As Table 3 shows, exact values of the maximal cardinality of a d-cap, C_d, are only known for dimension one to six [7].

Dimension	1	2	3	4	5	6	7
Maximal cardinality d-cap	2	4	9	20	45	112	?

Table 3: Known maximal cardinalities of d-caps.

1.3 Thesis Structure

In the following chapters the geometrical intepretation of the cap set problem in affine space will be considered. After an introduction to affine geometry and lower bounds, some proofs of known maximal cardinalities of caps in small dimensons are given. Furthermore, two recursive methods to obtain upper bounds on maximal capsizes in any dimension are described, extended, applied on dimensions seven to ten and compared.

2 About Affine Geometry

An affine space is nothing more than a vector space
whose origin we try to forget about,
by adding translations to the linear maps.
([1], page 32)

As became clear in Chapter 1, the cap set problem is defined in affine geometry. Therefore, it is important to understand the basics of affine geometry. In this chapter, the outlines of affine space and its geometry are introduced. It is assumed that the general definitions of (linear) algebra are known. The following formulations of definitions and remarks follow from [1, 7, 14].

Definition 2 (Affine subspace)**.** Let L be a linear subspace of vector space V and $p \in V$. The translation of L by p, i.e. $L + p = \{v + p \mid v \in L\}$, is an affine subspace.

Remarks.
Affine subspaces are also called flats.
Every vector space is an affine space, but not all affine spaces are vector spaces.
$\mathbb{F}_3^d$ is an affine space.
The affine subspaces of dimension zero are points, those of dimension one are lines and those of dimension two are planes.

Definition 3 (Affine hyperplane)**.** An affine hyperplane of a d-dimensional affine space is a $(d-1)$-dimensional affine subspace.

Remarks.
An affine hyperplane that contains the origin is a linear hyperplane.
An affine hyperplane of $\mathbb{F}_3^d$ has the form $\{x \in \mathbb{F}_3^d \mid x \cdot y = c\}$, for $y \in \mathbb{F}_3^d \backslash \{0\}$ fixed and $c \in \mathbb{F}_3$.

Definition 4 (Dimension)**.** The dimension of an affine space is the number of vectors in the basis of the corresponding vector space.

Definition 5 (Parallel subspaces)**.** Affine subspaces of the same dimension are parallel if they are translations of the same linear subspace.

Remark. $\mathbb{F}_3^d$ can be decomposed in three parallel affine hyperplanes, i.e.
$\mathbb{F}_3^d = \{x \in \mathbb{F}_3^d \mid x \cdot y = 0\} \cup \{x \in \mathbb{F}_3^d \mid x \cdot y = 1\} \cup \{x \in \mathbb{F}_3^d \mid x \cdot y = 2\}$, for $y \in \mathbb{F}_3^d \backslash \{0\}$ fixed.

Definition 6 (Affine independence)**.** The points $x_1, x_2, ..., x_k$ in a affine space are affinely independent if $\sum_{i=1}^k c_i x_i = 0$ with $\sum_{i=1}^k c_i = 0$ implies $c_1 = c_2 = \ldots = c_k = 0$.

Remark. Three points in $\mathbb{F}_3^d$ are collinear if they are affinely dependent. It follows that $a + b + c = 0$, what agrees with the statement in Chapter 1.

Definition 7 (Affine transformation)**.** An affine transformation is a bijection from an affine space to itself that preserves lines.

Remarks.
The dot product defined on $\mathbb{F}_3^d$ and translations are affine transformations.
An affine transformation preserves (hyper)planes and caps.

The following proposition on hyperplanes and lower dimensional subspaces will be applied in the hyperplane counting method in Chapter 5.

Proposition 1. *The number of hyperplanes containing a fixed k-dimensional subspace in $\mathbb{F}_3^d$ is equal to*

$$\frac{3^{d-k}-1}{2}.$$

Proof. Let S be a k-dimensional subspace of $\mathbb{F}_3^d$ which contains the origin. Then the natural map $\mathbb{F}_3^d \to \mathbb{F}_3^d/S \cong \mathbb{F}_3^{d-k}$ gives a bijection between hyperplanes of $\mathbb{F}_3^d$ containing S and hyperplanes of $\mathbb{F}_3^{d-k}$ containing the origin. With the hyperplanes containing the origin, they can be seen as linear subspaces of a vector space.

The hyperplanes of of $\mathbb{F}_3^{d-k}$ that contain the origin are determined by nonzero normal vector, of which there are $3^{d-k}-1$. Since there are two nonzero normal vectors determining each hyperplane, there are $\frac{3^{d-k}-1}{2}$ hyperplanes containing the origin.

Note that if the fixed subspace does not contain the origin, it can be translated. □

3 Lower Bounds on Maximal Caps

This chapter concerns the lower bound on the maximal cardinality of a cap in $\mathbb{F}_3^d$. Remember from Table 3 in Chapter 1 that for dimensions one to six the exact cardinalities of the maximal caps are known. Hence, for these dimensions the best known lower bound is equal to C_d. With this, the focus lies on establishing lower bounds for maximal capsizes in dimension seven to ten.

Two theorems will introduce constructions to obtain a cap in a dimension by using known caps in lower dimensions. Theorem 1 states a simplified version of the general product construction theorem first stated by Mukhopadhyay in [11] and reformulated by Edel and Bierbrauer [8, 2]. Theorem 2, the doubling construction, is a special case of the general product construction. Note that Theorem 2 does not follow from Theorem 1.

Theorem 1 (Product construction)**.** *Let $A \subset \mathbb{F}_3^{d_1}$ and $B \subset \mathbb{F}_3^{d_2}$ be caps. Then there is a cap in $\mathbb{F}_3^{d_1+d_2}$ of size $|A||B|$.*

The best lower bounds found with the product construction are displayed in Table 5 on page 6. To obtain the results, the known values of C_d for $d = 1, \dots, 6$ are used.

The doubling construction, as well as the general product construction theorem, relies besides affine spaces also on projective spaces. The definition by Cameron should give an idea of projective spaces [6].

Definition 8 (Projective space)**.** A projective space $PG(d, q)$ is defined by a $(d+1)$-dimensional vector space V over the field $\mathbb{F}_q$. The points, lines, planes, etc. are the subspaces of V of dimension one, two, three, etc.

Theorem 2 (Doubling construction)**.** *Let $A \subset PG(d, q)$. Then there is a cap in $\mathbb{F}_3^{d+1}$ of size* $2|A|$.

To obtain results with the doubling construction, capsizes in projective spaces need to be known. The best known lower bounds on the maximal cardinality of a cap in $PG(d, 3)$ are given in Table 4. These results are obtained from [9]. The outcome of the doubling construction based on the results in Table 4 are listed in Table 5.

Dimension	1	2	3	4	5	6	7	8	9	10
Lower bound	2	4	10	20	56	112	248	532	1120	2744

Table 4: Lower bounds on maximal capsizes in $PG(d, 3)$.

Dimension	2	3	4	5	6	7	8	9	10
Product construction	4	8	18	40	90	224	448	1008	2240
Doubling construction	4	8	20	40	112	224	496	1064	2240

Table 5: Lower bounds on C_d by product and doubling construction.

Clearly, the doubling construction yields better lower bounds than the product construction. This is mainly the case because there is more known about caps in $PG(d, 3)$ than about caps in $\mathbb{F}_3^d$. Therefore, the doubling construction is in general the best construction method for caps.

However, there is a better lower bound known for the maximum size of a 7-cap. In 1994, Calderbank and Fishburn constructed a 7-cap of size 236 [5]. The construction follows from a more involvied generalization of the general product construction [8]. Unfortunately, this new bound does not lead to an improvement on the other lower bounds.

Table 6 summarizes the best known lower bounds on C_d, including the equalities with C_d for the dimensions one to six.

Dimension	1	2	3	4	5	6	7	8	9	10
Lower bound	2	4	9	20	45	112	236	496	1064	2240

Table 6: Best known lower bounds on C_d.

4 Maximal Caps in Low Dimensions

As Table 3 in Chapter 1 shows, only for the dimensions one to six the exact value of the maximal cardinality of a cap is known. This is mainly because there are no patterns in either the construction of maximal caps or in the proofs. In this chapter the maximal caps in dimensions one, two and three willl be constructed and their optimality will be proven.

For the construction of the maximal caps, remember the SET cards from Chapter 1. The number of characteristics of the cards correspond to the dimension. Hence, leaving out one characteristic yields a representation of the three-dimensional $\mathbb{F}_3^3$, leaving out two characteristics $\mathbb{F}_3^2$ and leaving out three $\mathbb{F}_3$.

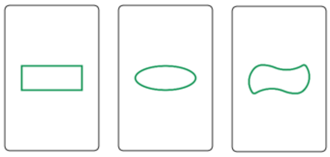

Figure 2: $\mathbb{F}_3$ in SET cards.

First, consider the three points in $\mathbb{F}_3$ as SET cards. In the representation in Figure 2, the characteristics number, shading and colour are fixed at one, empty and green, respectively. Hence, the only variable charactistic is shape. From the SET rule it is clear that the three cards form a SET. By definition, two cards cannot be a SET. Hence, the maximal cardinality of a 1-cap is two.

For the two-dimensional and three-dimensional caps both the construction and the proof are more involved. First, a maximal 3-cap and 4-cap are obtained. Then, to complete the proofs of Propositions 3 and 5, it is showed that no larger caps exist.

In Figure 3 $\mathbb{F}_3^2$ is represented by SET cards. The characteristics number and shading are fixed on one and empty. Figures 4 and 5 show the same 2-cap of size four. With either the SET rule for the cards or the geometrical SET rule it can be checked that these cards form a cap.

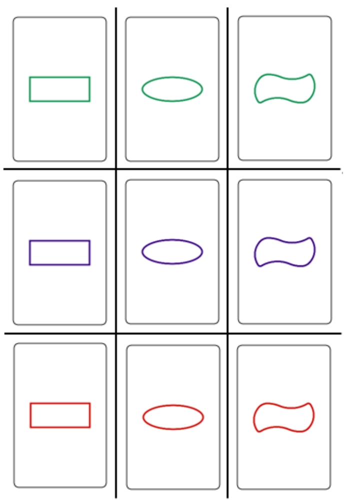

Figure 3: $\mathbb{F}_3^2$ in SET cards.

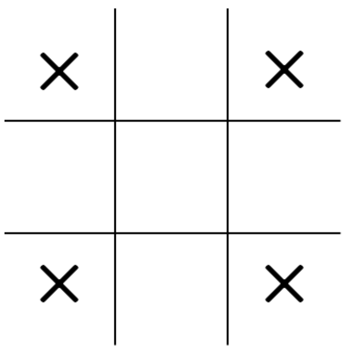

Figure 4: 2-cap of four cards, schematically.

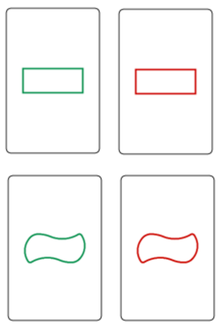

Figure 5: 2-cap of four cards.

In Figure 6 $\mathbb{F}_3^3$ is represented by SET cards. The characteristics number is fixed on one. Figures 7 and 8 show the same 3-cap of size nine. Again with the SET rule, it can be verified that the set of cards is indeed a cap.

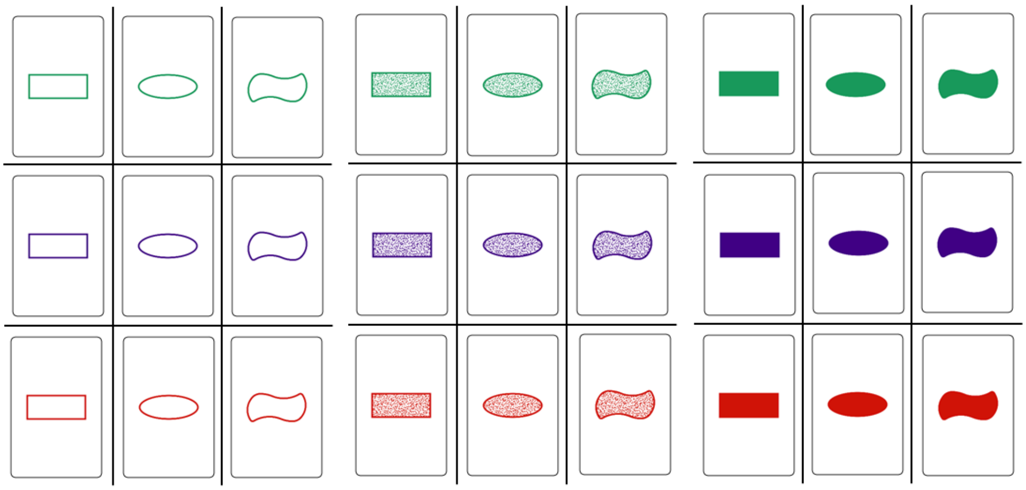

Figure 6: $\mathbb{F}_3^3$ in SET cards.

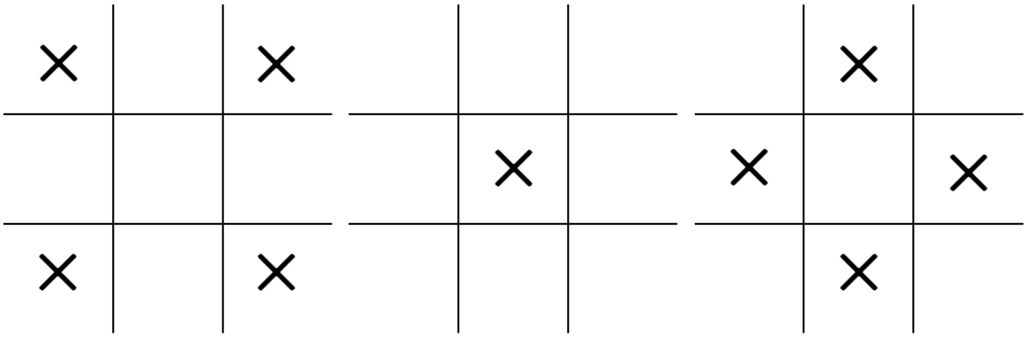

Figure 7: 3-cap of nine cards, schematically.

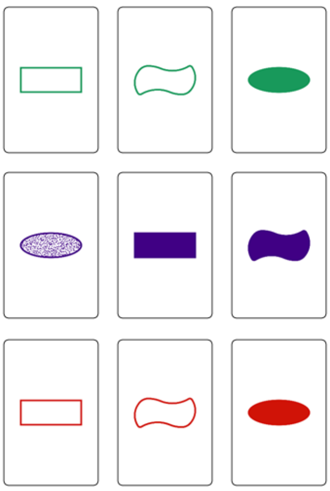

Figure 8: 3-cap of nine cards.

Now the caps of maximal size are constructed, it suffices to show that there exist no larger caps. Hence, the following proofs on the maximal size of 3-caps and 4-caps follow by contradiction on the assumption that a larger cap does exist. The contradiction is derived from the pigeonhole principle.

Proposition 2 (Pigeonhole principle)**.** *If n objects are distributed over m places with $n > m$, then there is a place receiving at least two objects.*

Proposition 3. *A maximal 2-cap has size four.*

Proof. In Figure 5 a 2-cap of size four is shown. The proof proceeds by condradiction. Assume there is a 2-cap A with five points: x_1, x_2, x_3, x_4 and x_5.

The plane $\mathbb{F}_3^2$ can be decomposed as a union of three horizontal lines as in Figure 9. Since A is cap, each line contains at most two points. This means that two lines contain two points each and one line one point, say x_5. Let H be the line containing x_5.

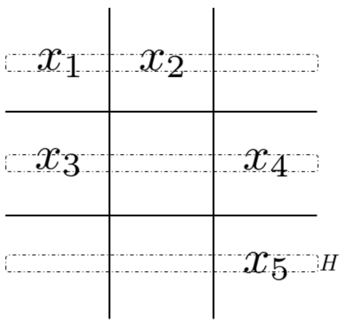

Figure 9: $\mathbb{F}_3^2$ as union of horizontal lines.

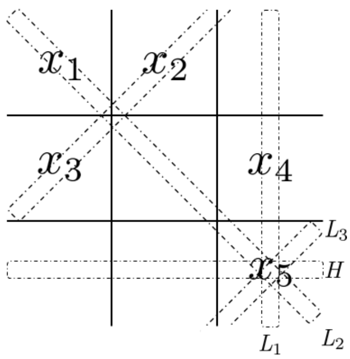

Figure 10: The four lines through x_5.

x_5 lies on exactly four lines in the plane: the horizontal line H, the vertical line L_1 and two diagonal lines L_2 and L_3. These lines cover each point in $\mathbb{F}_3^2$ not equal x_5 exactly once, as Figure 10 shows. Hence, the four points x_1, x_2, x_3 and x_4 all lie on these lines. Since H only contains x_5, the four points are distrubuted over the other three lines L_1, L_2 and L_3. By the pigeonhole principle, two of the four points lie on one of the three lines. Since this line also contains x_5, it now contains three points. Hence, they form a SET. This means that A is not a cap and hence, there exists no 2-cap of size five. □

Proposition 4. *A maximal 3-cap has size nine.*

Proof. In Figure 8 a 3-cap of size nine is shown. The proof proceeds by contradiction. Assume there is a 3-cap A that contains ten points.

$\mathbb{F}_3^3$ can be decomposed as a union of three parallel planes. The intersection of a plane and a 3-cap is a 2-cap. By Proposition 3, it contains at most four points. Therefore, there are either two planes containing four and one plane containing two points or one plane containing four and two planes containing three points. Let H be the plane containing the least number of points. Then there are at least seven points not contained in H.

Let a and b be two points in H. Then there are three other planes in $\mathbb{F}_3^3$ containing a and b, say P_1, P_2 and P_3. H, P_1, P_2 and P_3 cover each point in $\mathbb{F}_3^3$ not on the line through a and b exactly once. By the pigeonhole principle, at least one of the planes P_i must contain three of the seven points not contained in H. Hence, P_i contains five points. Therefore, A is not a cap by Proposition 3. Hence, there exists no 3-cap of size ten. □

5 Method 1: Counting Hyperplanes

In [7] Davis and Maclagan describe a method to determine whether a set of points in $\mathbb{F}_3^d$ could be be cap or not. This method relies on counting arguments concerning hyperplanes. Before elaborating on this method, some new terms are introduced.

Definition 9 (Hyperplane triple)**.** A hyperplane triple (ν_0, ν_1, ν_2) is a decomposition of a cap $A \subset \mathbb{F}_3^d$ over three parallel hyperplanes of $\mathbb{F}_3^d$, H_0, H_1 and H_2, where $\nu_c = |A \cap H_c|$, for $c \in \mathbb{F}_3$.

Definition 10 (n-Marked hyperplanes)**.** An n-marked hyperplane is a pair of the form $(H, \{x_1, ..., x_n\} \subset H \cap A\}$, with H a hyperplane and A a cap.

5.1 Method

To verify whether a set of points, A, in $\mathbb{F}_3^d$ are a cap or not the space is divided into three parallel hyperplanes of dimension $d-1: H_0, H_1$ and H_2. Note that $\cup_{c \in \mathbb{F}_3} H_c = \mathbb{F}_3^d$. Based on the data available on the maximal cardinality of a cap in $\mathbb{F}_3^{d-1}$, the possible hyperplane triples are listed. For every possible hyperplane triple, there is a variable $x_{\nu_0\nu_1\nu_2}$, which represents the number of (ν_0, ν_1, ν_2) hyperplane triples.

Next, an equation in the variables $x_{\nu_0\nu_1\nu_2}$ is obtained by counting the number of ways to decompose $\mathbb{F}_3^d$ as the union of three parallel planes. Additional equations are obtained by counting arguments on 2-marked and 3-marked hyperplanes. From these three (two in the two-dimensional case) equations solutions can be found. If there is a solution in which all $x_{\nu_0\nu_1\nu_2}$ are nonnegative integers, then there might be a cap in $\mathbb{F}_3^d$ of size $|A|$. If there is no such solution, there exists no cap of size $|A|$.

The method is applied in the proofs of Propositions 5 and 6 in Section 5.2. In dimensions three and four the exact value of the maximal d-cap is obtained. As will become clear in Section 5.3, this method gives only upper bounds for higher dimensions.

5.2 Maximal 3-Cap and 4-Cap

Remember that the SET cards are a visualization of $\mathbb{F}_3^4$. In Chapter 4 $\mathbb{F}_3^3$ was constructed with SET cards by leaving out the variations on number, see Figure 6 on page 9 . Also, a 3-cap consisting of twenty points was constructed, both schematically and in SET cards, see Figures 7 and 8 on page 9. The same constructions for $\mathbb{F}_3^4$ are visible in Figures 11, 12 and 13 on pages 12, 13 and 14.

To proof the maximal cardinalities of 3-caps and 4-caps are nine and twenty, respectively, the hyperplane counting method will be used. Since a 3-cap of size nine and a 4-cap of size twenty are constructed, it suffices to show there exists no 3-cap containing ten points and no 4-cap containing twenty-one points. Hence, the proofs follow by contradiction.

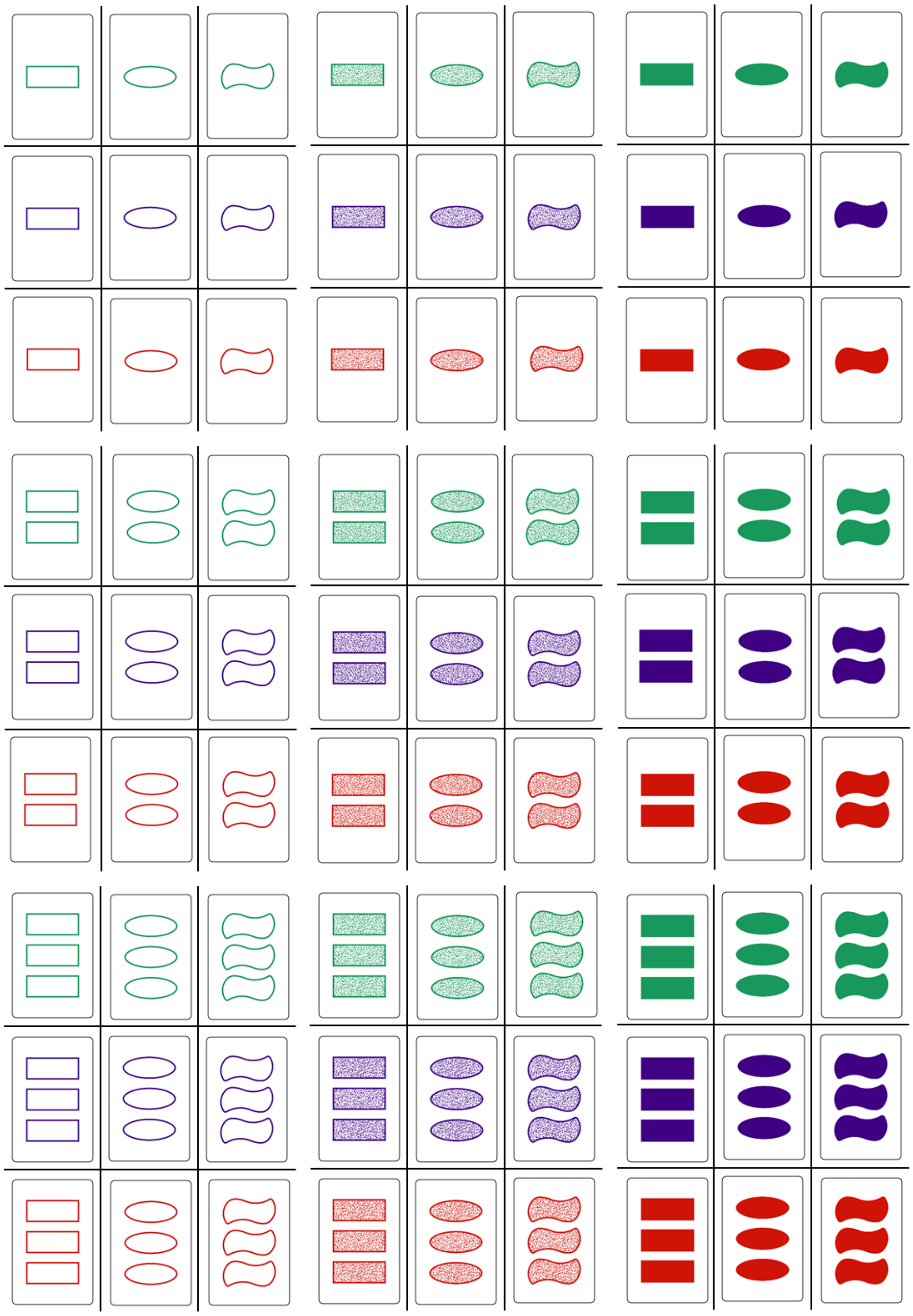

Figure 11: $\mathbb{F}_3^4$ in SET cards.

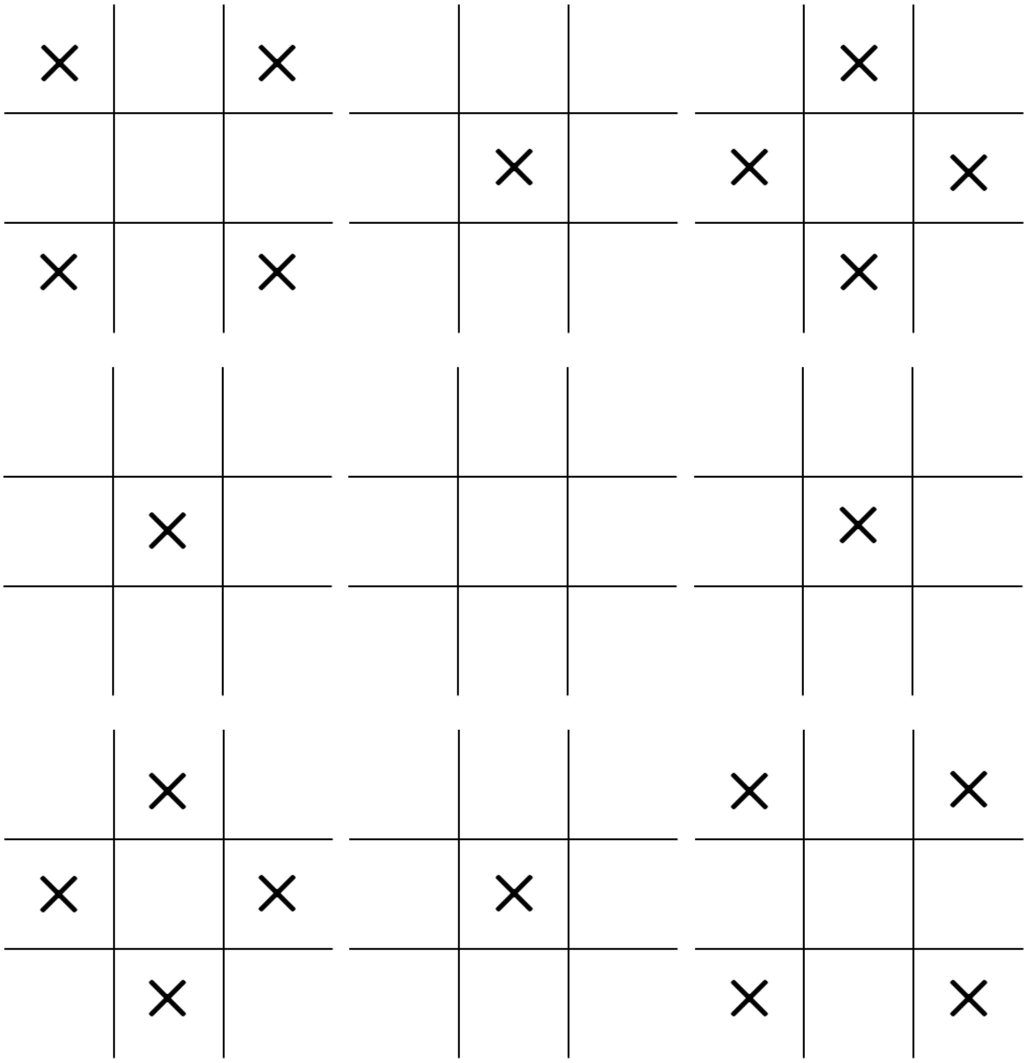

Figure 12: 4-cap of twenty points, schematically.

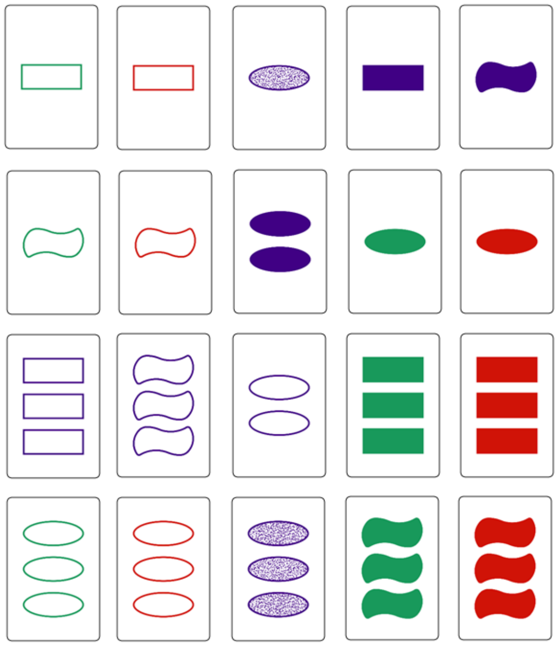

Figure 13: 4-cap of twenty cards.

Proposition 5. *A maximal 3-cap has nine points.*

Proof. A 3-cap of size nine has been constructed in Figure 8. The proof proceeds by contradiction. Assume there exists a 3-cap A with ten points.

The three dimensional $\mathbb{F}_3^3$ can be decomposed as a union of three parallel planes. Since a maximal 2-cap has 4 points, there are only two possible hyperplane triples: $(4,4,2)$ and $(4,3,3)$. Let x_{442} be the number of $(4,4,2)$ hyperplane triples and x_{433} the number of $(4,3,3)$ hyperplane triples.

The number of ways to decompose $\mathbb{F}_3^3$ as the union of three planes can be obtained in two ways. On one hand, there are $x_{4442}+x_{433}$ ways. On the other hand, there is a unique line through the origin of $\mathbb{F}_3^3$ perpendicular to each set of three parallel planes. These lines can be counted as follows. Each line through the origin is determined by a pair of nonzero points a and $-a$, of which there are $\frac{3^3-1}{2}=\frac{26}{2}=13$. This corresponds to Proposition 1. with $k=1$. Hence,

$$x_{442}+x_{433}=13. \tag{5.1}$$

To obtain a second equation for x_{442} and x_{433}, 2-marked (hyper)planes are introduced. By Proposition 1 in Chapter 2, there are four planes that contain a fixed pair of distinct points and hence, $4\binom{10}{2}=180$ 2-marked planes. On the other hand, for each $(4,4,2)$ hyperplane triple the number of 2-marked planes is $\binom{4}{2}+\binom{4}{2}+\binom{2}{2}=13$ and for each $(4,3,3)$ hyperplane triple $\binom{4}{2}+\binom{3}{2}+\binom{3}{2}=12$. Hence,

$$13x_{442}+12x_{433}=180. \tag{5.2}$$

Equations (5.1) and (5.2) resolve in a unique solution: $x_{442}=24$ and $x_{433}=-11$. Since x_{433} is negative, it contradicts the definition of x_{433}. Hence, there is no 3-cap with ten points. $\square$

Proposition 6. *A maximal 4-cap has twenty points.*

Proof. A 4-cap of size twenty has been constructed in Figure (12 or 13). The proof proceeds by contradiction. Assume there is a cap $A \subset \mathbb{F}_3^4$ with twenty-one points.

The four-dimensional $\mathbb{F}_3^4$ can be decomposed as the union of three parallel three-dimensional hyperplanes. Let x_{ijk} be the number of (i, j, k) hyperplane triples of A. Since the maximal cardinality of a 3-cap is nine by Proposition 5,

$$(i, j, k) \in \{(9, 9, 3), (9, 8, 4), (9, 7, 5), (9, 6, 6), (8, 8, 5), (8, 7, 6), (7, 7, 7)\}.$$

The number of possible decompositions of $\mathbb{F}_3^4$ as the union of three parallel hyperplanes equals both $\sum_{(i,j,k)} x_{ijk}$ and the number of lines through the origin of $\mathbb{F}_3^4$, which is equal to $\frac{3^4-1}{2} = 40$. Hence,

$$\sum_{(i,j,k)} x_{ijk} = 40 \tag{5.3}$$

A second equation can be obtained by counting 2-marked hyperplanes. By Proposition 1, the number of hyperplanes containing a fixed pair of distinct points, or a line, equals $\frac{3^{4-1}-1}{2} = 13$. Hence, there are $13\binom{21}{2} = 2730$ 2-marked hyperplanes.

In the same way as in the proof of Proposition5, the number of 2-marked hyperplanes equals

$$\left[\binom{9}{2} + \binom{9}{2} + \binom{3}{2}\right] x_{993} + \cdots + \left[\binom{7}{2} + \binom{7}{2} + \binom{7}{2}\right] x_{777}.$$

Computing the coefficients gives the following equation.

$$75x_{993} + 70x_{984} + 67x_{975} + 66x_{966} + 66x_{885} + 64x_{876} + 63x_{777} = 2730. \tag{5.4}$$

By counting 3-marked hyperplanes, a third equation can be obtained. Note that three points in a cap cannot be collinear. Hence, the three points span a plane. By Proposition 1, the number of hyperplanes containing a fixed plane equals $\frac{3^{4-2}-1}{2} = 4$. Hence, there are $4\binom{21}{3} = 5320$ 3-marked hyperplanes. Following the count of 2-marked hyperplanes yields

$$169x_{993} + 144x_{984} + 129x_{975} + 124x_{966} + 122x_{885} + 111x_{876} + 105x_{777} = 5320. \tag{5.5}$$

Three equations have been obtained for seven variables, which in principle yields infinitely many solutions. Fortunately, the requirement that the variables are nonnegative integers suffices for this proof. Adding 693 times Equation (5.3) to three times Equation (5.5) and then subtracting six times Equation (5.4), yields

$$5x_{984} + 8x_{975} + 9x_{966} + 3x_{885} + 2x_{876} = 0.$$

The only nonnegative solution of this equation is $x_{984} = x_{975} = x_{975} = x_{966} = x_{876} = 0$. Subtracting 63 times Equation (5.3) from Equation (5.4) yields

$$12x_{993} + 7x_{984} + 4x_{97} + 3x_{996} + 3x_{88} + x_{876} = 210.$$

This combines to $12x_{993} = 210$, which gives a non-integer solution. Hence, there is no 4-cap with twenty-one points. □

5.3 Implementation

After applying the hyperplane counting method on three-dimensional and four-dimensional caps, the method will be extended to higher dimensions.

In theory, there already are infinitely many solutions in the four-dimensional case. Hence, it is no surprise that this method does not yield exact answers for dimension five and higher. However, upper bounds on the maximal cardinality of d-caps can be obtained.

Since the number of possible hyperplane triples and therewith the number of variables increase drastically in higher dimensions, the method is implemented as a mixed integer linear program (MILP) in both Python[1] and MATLAB[2]. Whereas MATLAB has its own MILP solver, intlinprog, Python requires an external solver such as Gurobi[3].

The implementation of the method is divided over a few files. Figure 14 shows a diagram of the different code files with a short description and their interaction. All the code is included in Appendices A.1 and A.2.

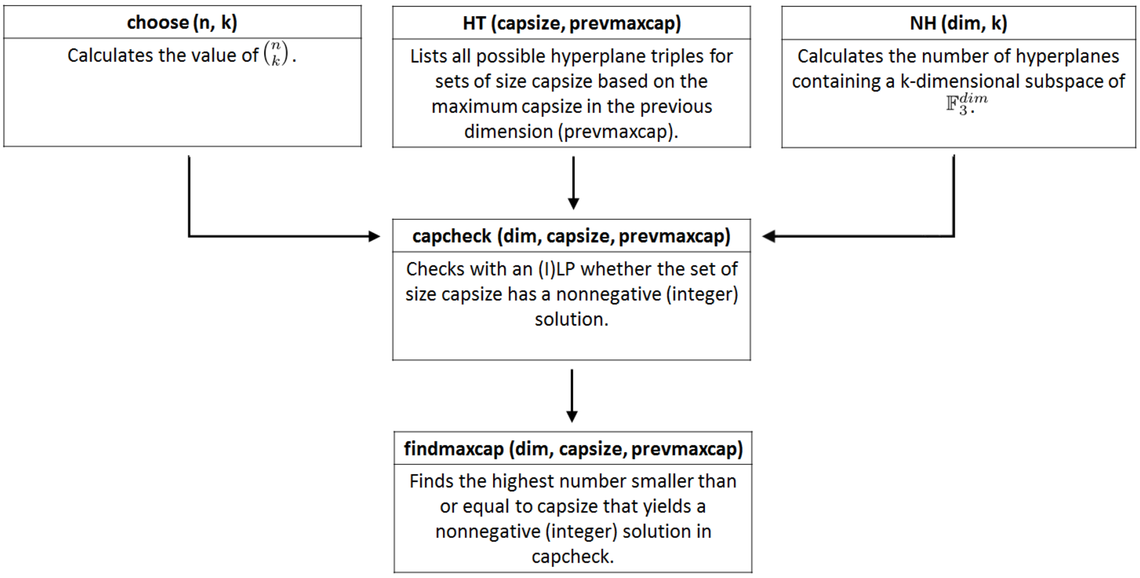

Figure 14: Diagram of code files.

The ILP grows fast and therewith the running time already exceeds the capacity of a personal computer by dimension nine. Therefore, the ILP for dimensions nine and ten are solved on the NEOS Server[4], which offers the Gurobi solver for MILP's as MPS files.

On the other hand, the LP is considered. This results in programs small enough to run without an external server.

The results of the different implementations are shown in Table 7. In all cases the best known upper bound of the maximal capsize in the previous dimension is used. That is, either the known maximal cardinality of a cap or the bound obtained with the ILP.

[1] https://www.python.org/
[2] https://www.mathworks.com/products/matlab.html
[3] http://www.gurobi.com/
[4] https://neos-server.org/

Dimension	1	2	3	4	5	6	7	8	9	10
Known max cap	2	4	9	20	45	112	-	-	-	-
Python - ILP	-	4	9	20	48	114	291	771	2070	5619
Python - LP	-	4	9	21	48	114	292	771	2070	5619
MATLAB - ILP	-	4	9	20	48	114	291	771	2070	-
MATLAB - LP	-	4	9	21	48	114	292	771	2070	5619

Table 7: Results of implementation of Method 1.

As Table 7 shows, the Python and MATLAB programs lead to the same upper bounds, as expected. Furthermore, the ILP and the LP yield results that barely differ. Since there is no difference in the dimensions eight, nine and ten, the integer constraints do not seem to pay for higher dimensions.

5.4 4-Marked Hyperplanes

A possibility to improve the hyperplane counting method is adding a constraint based on 4-marked hyperplanes. With the 3-marked hyperplanes it was clear that three points define a plane, because three points can not lie on a line in a cap.

However, four points can determine either a three-dimensional subspace or a two-dimensional subspace, a plane. Therefore, the constraint based on 4-marked hyperplanes requires an extra variable, depending on the number of possibilities that four points, a quartet, are degenerated.

Definition 11 (Degenerated quartet)**.** A quartet of points is degenerated if the points define a two-dimensional subspace instead of a three-dimensional subspace.

Remark. The four points in a degenerated quartet are affinely dependent.

Let t be the number of degenerated quartets in a cap $A \subset \mathbb{F}_3^d$. With this, t is a nonnegative integer smaller or equal to the number of possible quartets in A:

$$0 \leq t \leq \binom{|A|}{4}.$$

Both the upper and lower bounds on t will be improved in this section.

For the upper bound on t, the binary constant weight codes are consider. In general, $A(n, \delta, w)$ is the maximum size of a binary code with word length n, minimum distance d and weight w. To compare with degenerated quartets, the mimimum distance and weight have to be fixed on four. The weight is four because quartets of points are considered. The minimum distance is four because two quartets can have at most two points in commom (with three points in common there would lie five points in a plane, which contradicts being a cap) and hence the symmetric difference is four, six or eight. Therefore, $A(n, 4, 4)$ with n the size of the (presumed) cap ($|A|$) is considered. From [4] the following theorem on $A(n, 4, 4)$ is obtained.

Theorem 3. *Let $A(n, 4, 4)$ be the maximum size of a binary constant weight code with word length n, minimum distance 4 and weight 4. Then*

$$24A(n,4,4) = \begin{cases} n(n-1)(n-2) & \text{if } n \equiv 2 \text{ or } 4 \text{ mod } 6, \\ n(n-1)(n-3) & \text{if } n \equiv 1 \text{ or } 3 \text{ mod } 6, \\ n(n^2-3n-6) & \text{if } n \equiv 0 \text{ mod } 6. \end{cases}$$

In 2006 the case $n \equiv 5 \bmod 6$ was solved, see [10]. Combining Theorem 3 with [10] for the specific cases of $|A| = 47$ and $|A| = 113$ yields the values of $A(|A|, 4, 4)$ displayed in Table 8.

Dimension	4	4	5	5	5	5	6	6	6
Capsize	21	20	48	47	46	45	114	113	112
Upper bound	315	285	4308	3959	3795	3465	60078	57997	56980

Dimension	7	7	8	9	10
Capsize	291	290	771	2070	5619
Upper bound	1026745	1012680	18997440	369036495	7388120142

Table 8: Upper bounds on t by dimension and capsize.

The improvement of the lower bounds is based on an explicit search for optimal values of t. The used code, small extensions on the code of Section 5.3, can be found in Appendix A.3. Below, only the results for dimenson seven are illustrated.

The graph in Figure 15 describes the relation between the number of degenerated quartets and the upper bound on the maximal size of a 7-cap. In the graph two horizontal lines are marked: $y = 112$ and $y = 236$. The first line indicates the absolute lower bound of the function, since 112 is the maximal size of a 6-cap. The second line indicates the best known lower bound on the maximal size of a 7-cap [5]. This means that the graph, and hence the upper bound on the maximal size of a 7-cap, cannot be lower than 236. The two vertical lines, $t = 185000$ and $t = 451260$ indicate the bounds on the number of degenerated quartets to satisfy this condition. Taking a close look at the graph, see Figure 16, results in excluding of the values 44708 to 448492 from the possible numbers of degenerated quartets. Hence, $185000 \leq t \leq 44707$ or $448493 \leq t \leq 451260$.

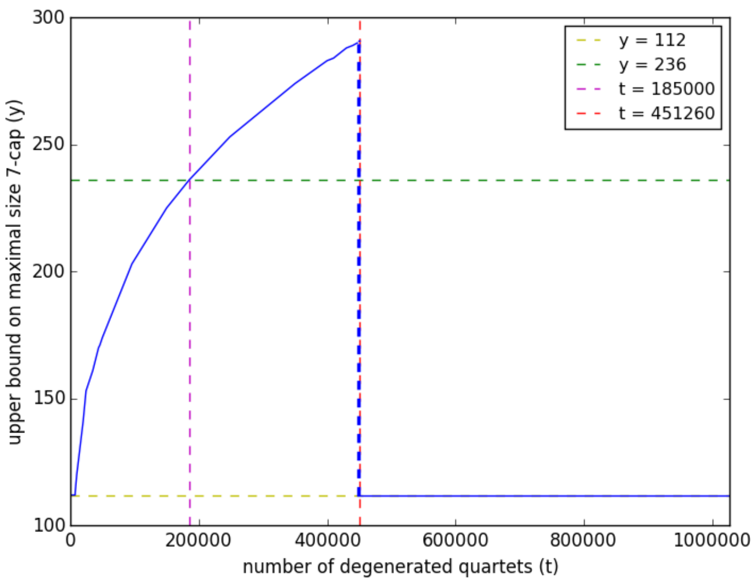

Figure 15: Upper bounds on C_7 for varying t.

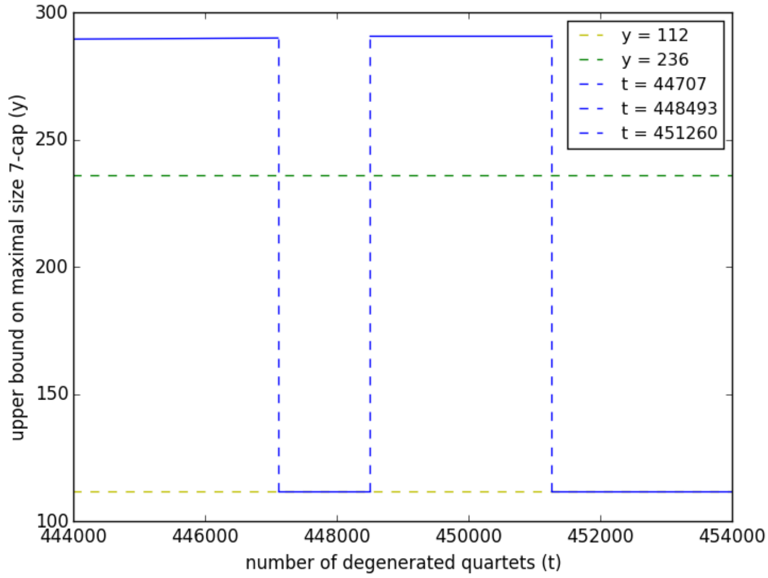

Figure 16: Close-up upper bounds on C_7 for varying t.

The maximum upper bound, 291, is only reached when $448493 \leq t \leq 451260$. This bound is one less than the upper bound obtained by the LP and herewith equal to the result of the ILP.

Equivalent steps for dimensions four to nine lead to the results in Table 9. The graphs for dimensions four, five, six and eight are included in Appendix A.4. Both the jumps in the graph and the equivalence of the maximum upper bound with the result of the ILP are consistent with the results in these dimensions.

For the bounds on the number of degenerated quartets there is looked into both the values for which the maximum upper bound on the maximum size of a d-cap is obtained and which values are possible on acccount of the lower bounds on the maximum size of a d-cap established in Chapter 3. Since the values for which the maximum upper bound is obtained (1) are included in the possible values based on the lower bounds as an isolated interval, only the other interval is written down as case (2). Note that for dimension four there is no second interval since the lower bound on C_4 is equal to the obtained maximum upper bounds. Furthermore, for dimension nine the program takes too much time to calculate the bounds of the second interval.

Dimension	4	5	6	7	8	9
Maximal upper bound	20	48	114	291	771	2070
Lower bound t (1)	206	2710	30533	448493	7462603	129107939
Upper bound t	227	2809	30871	451260	7465547	129170830
Lower bound t (2)	-	1980	28261	185000	1181270	?
Upper bound t (2)	-	2651	30142	44707	7440062	?

Table 9: Maximal upper bounds on C_d and corresponding bounds on t.

The explicit search to lower bounds have not only improved the lower bounds on the maximal size of a d-cap, but also the upper bound. Hence, the upper bounds on t in Table 9 can be compared to the upper bounds in Table 8, obtained with expressions for $A(n, 4, 4)$. The difference between the bounds rows rapidly to the advantage of the explicit search. However, practically all values of $A(n, 4, 4)$ are known, while the better upper bounds require an extensive search which becomes impracticable for higher dimensions.

5.5 Results

In Table 10 the results of the hyperplane counting method are aggregated. The upper bounds on the maximal size of a d-cap obtained with the ILP, LP and the addition of the constraint based on 4-marked hyperplanes are stated to compare to each other and to the known maximal capsizes for the dimensions one to six.

Dimension	1	2	3	4	5	6	7	8	9	10
Known max cap	2	4	9	20	45	112	-	-	-	-
Upper bound - ILP	-	4	9	20	48	114	291	771	2070	5619
Upper bound - LP	-	4	9	21	48	114	292	771	2070	5619
Upper bound - 4-marked	-	-	-	20	48	114	291	771	2070	-

Table 10: Upper bounds by ILP, LP and 4-marked hyerplanes.

The difference between the results of the three variants of the method are minimal. Especially for the higher dimension, while the difference in running time increases rapidly because the ILP grows the fastest. Nevertheless, the LP and the program that includes the 4-marked hyperplane constraint also grow too much to be extendable to higher dimensions.

6 Method 2: Fourier Transform

In [3] Bierbrauer and Edel describe a method to obtain upper bounds for the density of caps in the d-dimensional space over the field with q elements. This density is expressed as the following ratio:

$$c_d(q) = \frac{C_d(q)}{|\mathbb{F}_q^d|} = \frac{C_d(q)}{q^d},$$

with $c_d(q)$ the density and $C_d(q)$ the maximal cardinality of a cap in $\mathbb{F}_q^d$.

This method is based on the Fourier transform, as will become clear in Section 6.1. The main result is the following theorem.

Theorem 4. *Let $q > 2$ be a prime power. If $d \geq 3$, then*

$$c_d(q) \leq \frac{q^{-d} + c_{d-1}(q)}{1 + c_{d-1}(q)}.$$

Since the focus of this thesis lies on the specific case $q = 3$, Theorem 4 is reformulated to describe this case:

Theorem 5. *If $d \geq 2$, then*

$$c_d \leq \frac{3^{-d} + c_{d-1}}{1 + c_{d-1}}.$$

Remark 5.1. Theorem 4 does not hold in $\mathbb{F}_q^2$ for all q, but it does for $q = 3$.

In the following section, the proof of Theorem 5 will be given. This proof follows the proof of Theorem 4 by Bierbrauer and Edel, but will elaborate it for the case $q = 3$.

Subsequently, Theorem 5 will be applied to the dimensions two to ten to obtain upper bounds for the maximal cardinality of d-caps.

6.1 Proof of Theorem 5

Let $d \geq 2$ and $A \subset \mathbb{F}_3^d$ a cap. Let $x \cdot y$ be the dot product defined on $\mathbb{F}_3^d$. Let ζ be a complex primitive 3^{rd} root of unity and consider the complex number $U_y = \sum_{a \in A} \zeta^{a \cdot y}$. Define the real vector $\mathbf{u}$ of length $3^d - 1$ whose entries correspond to $|U_y|$, for $y \neq 0$.

Lemma 1. *Let $y \in \mathbb{F}_3^d \backslash \{0\}$. Then*

$$|U_y| \leq 3C_{d-1} - |A| = 3^d c_{d-1} - |A|.$$

Proof. Let $\nu_c = |\{a \in A \mid a \cdot y = c\}|$, for $c \in \mathbb{F}_3$. The set $\{v \in \mathbb{F}_3^d \mid v \cdot y = c\}$ forms a $(d-1)$-dimensional affine hyperplane of $\mathbb{F}_3^d$. Hence, $\nu_c \leq C_{d-1}$. Using the definition of U_y it follows that

$$|U_y| = |\sum_{a \in A} \zeta^{a \cdot y}| = |\sum_{c \in \mathbb{F}_3} \nu_c \zeta^c|. \tag{6.1}$$

Since $\sum_{c \in \mathbb{F}_3} \zeta^c = 0$, $\sum_{c \in \mathbb{F}_3} C_{d-1} \zeta^c = 0$. Hence,

$$|\sum_{c \in \mathbb{F}_3} \nu_c \zeta^c| = |\sum_{c \in \mathbb{F}_3} -\nu_c \zeta^c| = |\sum_{c \in \mathbb{F}_3} (C_{d-1} - \nu_c) \zeta^c|. \tag{6.2}$$

Combing Equations (6.1) and (6.2) with the triangle inequality yields

$$|U_y| = |\sum_{c \in \mathbb{F}_3} (C_{d-1} - \nu_c) \zeta^c| \leq \sum_{c \in \mathbb{F}_3} |C_{d-1} - \nu_c||\zeta^c|. \tag{6.3}$$

Because $|\zeta^c| = 1$ and $\nu_c \leq C_{d-1}$, it follows that

$$\sum_{c\in\mathbb{F}_3} |C_{d-1} - \nu_c||\zeta^c| = \sum_{c\in\mathbb{F}_3} |C_{d-1} - \nu_c| = \sum_{c\in\mathbb{F}_3} C_{d-1} - \nu_c. \tag{6.4}$$

Hence, by Equations (6.3) and (6.4),

$$|U_y| \leq \sum_{c\in\mathbb{F}_3} C_{d-1} - \nu_c = 3C_{d-1} - |A|.$$

□

Lemma 2. $||\mathbf{u}||^2 = |A|(3^d - |A|)$.

Proof. Using that $U_0 = \sum_{a\in A} \zeta^{a\cdot 0} = |A|$ and the definition of the norm of a complex vector, it follows that

$$\begin{aligned}
||\mathbf{u}||^2 &= \sum_{y\in\mathbb{F}_3^d\setminus\{0\}} U_y\overline{U_y} \\
&= \sum_{y\in\mathbb{F}_3^d} U_y\overline{U_y} - U_0\overline{U_0} \\
&= \sum_{y\in\mathbb{F}_3^d} U_y\overline{U_y} - |A|^2 \\
&= \sum_{y\in\mathbb{F}_3^d} (\sum_{a\in A} \zeta^{a\cdot y})(\overline{\sum_{a\in A} \zeta^{a\cdot y}}) - |A|^2 \\
&= \sum_{y\in\mathbb{F}_3^d} (\sum_{a\in A} \zeta^{a\cdot y})(\sum_{a\in A} \zeta^{-a\cdot y}) - |A|^2 \\
&= \sum_{y\in\mathbb{F}_3^d} \sum_{a,b\in A} \zeta^{(a-b)\cdot y} - |A|^2.
\end{aligned} \tag{6.5}$$

If $a - b \neq 0$, then

$$\sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot y} = \sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot(y+e_j)} = \sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot y}\zeta^{(a-b)\cdot e_j}, \tag{6.6}$$

with e_j a standard unit vector and j a coordinate in which $a-b$ is nonzero. Then $\zeta^{(a-b)\cdot e_j} \neq 1$. Hence, Equation (6.6) can only be true when both sides equal zero. Therefore,

$$\sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot y} = 0. \tag{6.7}$$

If $a - b = 0$, then

$$\sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot y} = |\mathbb{F}_3^d| = 3^d. \tag{6.8}$$

If $a - b = 0$, then $a = b$. Combining this with Equations (6.7) and (6.8) yields

$$\sum_{y\in\mathbb{F}_3^d} \sum_{a,b\in A} \zeta^{(a-b)\cdot y} = \sum_{a,b\in A} \sum_{y\in\mathbb{F}_3^d} \zeta^{(a-b)\cdot y} = \sum_{a\in A} 3^d = |A|3^d. \tag{6.9}$$

Combining Equations (6.5) and (6.9) yields the desired result:

$$||\mathbf{u}||^2 = |A|3^d - |A|^2 = |A|(3^d - |A|).$$

□

For the following steps in the proof, another complex number is considered. Keeping in mind that A is a cap and hence $a_1 + a_2 + a_3 = 0$ implies $a_1 = a_2 = a_3 \; \forall a_1, a_2, a_3 \in \mathbb{F}_3^d$, the complex number S is defined as follows.

$$S = \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} \sum_{a_1, a_2, a_3 \in A} \zeta^{(\sum_{i=1}^3 a_i) \cdot y}.$$

Lemma 3. $|S| \leq \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} |U_y|^3$.

Proof. Using the definition of S, the triangle inequality and the definition of U_y, it follows that

$$\begin{aligned}
|S| &= |\sum_{y \in \mathbb{F}_3^d \setminus \{0\}} \sum_{a_1, a_2, a_3 \in A} \zeta^{(\sum_{i=1}^3 a_i) \cdot y}| \\
&\leq \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} | \sum_{a_1, a_2, a_3 \in A} \zeta^{(\sum_{i=1}^3 a_i) \cdot y}| \\
&= \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} | \sum_{a_1 \in A} \zeta^{a_1 \cdot y} \sum_{a_2 \in A} \zeta^{a_2 \cdot y} \sum_{a_3 \in A} \zeta^{a_3 \cdot y}| \\
&= \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} | \sum_{a_1 \in A} \zeta^{a_1 \cdot y}| \, | \sum_{a_2 \in A} \zeta^{a_2 \cdot y}| \, | \sum_{a_3 \in A} \zeta^{a_3 \cdot y}| \\
&= \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} |U_y|^3.
\end{aligned}$$

□

The definiton of U_y is closely related to the number of lines a subset of $\mathbb{F}_3^d$ of a certain size contains. This is explained in Proposition 7.

Proposition 7. *Let $B \subset \mathbb{F}_3^d$ that contains l lines. Then*

$$|B| + 6l = \frac{1}{3^d} \sum_{y \in \mathbb{F}_3^d} (U_y)^3.$$

Proof. Using the definition of U_y, it follows that

$$\begin{aligned}
\sum_{y \in \mathbb{F}_3^d} (U_y)^3 &= \sum_{y \in \mathbb{F}_3^d} (\sum_{a \in A} \zeta^{a \cdot y})^3 \\
&= \sum_{y \in \mathbb{F}_3^d} (\sum_{a_1 \in A} \zeta^{a_1 \cdot y} \sum_{a_2 \in A} \zeta^{a_2 \cdot y} \sum_{a_3 \in A} \zeta^{a_3 \cdot y}) \\
&= \sum_{y \in \mathbb{F}_3^d} \sum_{a_1, a_2, a_3 \in A} \zeta^{\sum_{i=1}^3 a_i \cdot y}. \qquad (6.10)
\end{aligned}$$

From the proof of Lemma 2 it is clear that $\sum_{y \in \mathbb{F}_3^d} \zeta^{(\sum_{i=1}^3 a_i) \cdot y} = 0$ whenever $\sum_{i=1}^3 a_i \neq 0$. Hence, assume $\sum_{i=1}^3 a_i = 0$. Then either $a_1 = a_2 = a_3$ or a_1, a_2 and a_3 lie on a line. If $a_1 = a_2 = a_3$, then

$$\sum_{y \in \mathbb{F}_3^d} \zeta^{\sum_{i=1}^3 a_i \cdot y} = \sum_{y \in \mathbb{F}_3^d} \zeta^{0 \cdot y} = 3^d,$$

for a total contribution of $|A|3^d$.

If a_1, a_2 and a_3 define a line, then that line is counted six (3!) times because of the possible orderings. Hence,

$$\sum_{y \in \mathbb{F}_3^d} \zeta^{\sum_{i=1}^3 a_i \cdot y} = 3^d,$$

obtaining a contribution of $3^d 6l$.
Combining the results of the different cases with Equation 6.10 yields

$$\sum_{y \in \mathbb{F}_3^d} (U_y)^3 = 3^d(|A| + 6l).$$

Hence,

$$\frac{1}{3^d} \sum_{y \in \mathbb{F}_3^d} (U_y)^3 = |A| + 6l. \qquad \square$$

Lemma 4. $S = |A|(3^d - |A|^2)$.

Proof. A is a cap and therefore contains no lines. With this, the proof follows from Proposition 7. In the rewriting, the definitions of U_y and S are used.

$$\begin{aligned}
|A| &= \frac{1}{3^d} \sum_{y \in F} (U_y)^3 \\
&= \frac{1}{3^d} \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} (U_y)^3 + \frac{1}{3^d}(U_0)^3 \\
&= \frac{1}{3^d} \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} (\sum_{a \in A} \zeta^{a \cdot y})^3 + \frac{1}{3^d}|A|^3 \\
&= \frac{1}{3^d} \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} \sum_{a_1, a_2, a_3 \in A} \zeta^{\sum_{i=1}^3 a_i \cdot y} + \frac{1}{3^d}|A|^3 \\
&= \frac{1}{3^d} S + \frac{1}{3^d}|A|^3.
\end{aligned}$$

Hence,

$$S = |A|3^d - |A|^3 = |A|(3^d - |A|^2). \qquad \square$$

To obtain the statement of Theorem 5 all lemmas proven above will be used.
Applying Lemma 1 on the first u_y in the result of Lemma 3 yields

$$|S| = \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} |U_y|^3 \;\leq\; (3^d c_{d-1} - |A|) \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} |U_y|^2 \tag{6.11}$$

Applying the definition of the norm of a real vector and Lemma 2 to Equation (6.11) yields

$$|S| \leq (3^d c_{d-1} - |A|)||\mathbf{u}||^2 = (3^d c_{d-1} - |A|)|A|(3^d - |A|). \tag{6.12}$$

Let $|A|$ be C_d. Then Equation (6.12) gives

$$\begin{aligned}
|S| &\leq (3^d c_{d-1} - C_d)C_d(3^d - C_d) \\
&= (3^d c_{d-1} - 3^d c_d)3^d c_d(3^d - 3^d c_d) \\
&= 3^{3d} c_d(c_{d-1} - c_{d-1}c_d - c_d + c_d^2).
\end{aligned} \tag{6.13}$$

$\{0,1\}^d \subset \mathbb{F}_3^d$ is a cap. Hence,

$$C_d \geq |\{0,1\}^d| = 2^d > \sqrt{3}^d = \sqrt{3^d}.$$

With $|A|$ equal to C_d, Lemma 4 yields

$$S = C_d(3^d - C_d^2) < 0.$$

Hence,

$$|S| = -S = -C_d(3^d - C_d^2) = -3^d c_d(3^d - 3^{2d}c_d^2) = -3^{2d}c_d - 3^{3d}c_d^3. \tag{6.14}$$

Combining Equations (6.13) and (6.14) yields

$$3^{3d}c_d(c_{d-1} - c_{d-1}c_d - c_d + c_d^2) \geq -3^{2d}c_d - 3^{3d}c_d^3.$$

Dividing by $-3^{3d}c_d$ and further rewriting yields

$$c_d(1 + c_{d-1}) \leq 3^{-d} + c_{d-1}.$$

Hence, the final result is obtained:

$$c_d \leq \frac{3^{-d} + c_{d-1}}{1 + c_{d-1}}.$$

6.2 Application

To obtain upper bounds for the maximal cardinality of caps in $\mathbb{F}_3^d$, Theorem 5 needs to be reinterpreted for C_d instead of c_d. Since $c_d = \frac{C_d}{3^d}$, the rewriting is easily done. It results in Corollary 5.1.

Corollary 5.1. *If $d \geq 2$, then*

$$C_d \leq \frac{1 + 3C_{d-1}}{1 + 3^{-d+1}C_{d-1}}.$$

Now, an upper bound for the maximal cardinality of caps can be obtained for specific dimensions. The Python code used to obtain the results is included in Appendix B.

Table 11 displays three kinds of upper bounds, obtained in three different ways, depending on the value of C_{d-1}: (1) only obtained by Corollary 5.1 itself, (2) obtained by the known maximal size for the first six dimensions and the therewith obtained results and (3) obtained by the best known upper bound, i.e. the known maximal size or the bound obtained by the ILP in Section 5.3.

Dimension	1	2	3	4	5	6	7	8	9	10
Known max cap	2	4	9	20	45	112	-	-	-	-
Upper bound (1)	-	4	9	21	50	125	320	838	2230	6010
Upper bound (2)	-	4	9	21	48	114	292	773	2075	5632
Upper bound (3)	-	4	9	21	48	114	292	771	2070	5619

Table 11: Results of Corollary 5.1 in lower dimensions, depending on the values of C_{d-1}.

There are two main observations based on the results. Firstly, they show that a small difference in the upper bound on a cap in a certain dimension causes a larger difference in the following dimension. This difference increases rapidly in higher dimensions. Secondly, the results that uses the best known upper bound are equal to the results from the LP in Section 5.3, which uses the same upper bounds for C_{d-1}.

7 Considering the Methods

In this chapter the two methods from Chapters 5 and 6 will be compared and connected in a couple of different ways. First, there will be looked back at the degenerated quartets from the 4-marked hyperplanes of Section 5.4 with the knowledge from the proofs in Section 6.1. Subsequently, the complex number U_y and vector $\mathbf{u}$ from Section 6.1 willl be associated with the hyperplane triples and the solution from the ILP from Chapter 5.

7.1 A Review on Degenerated Quartets

In Section 5.4 upper and lower bounds on the number of degenerated quartets in $\mathbb{F}_3^d$ were established. The bounds followed from an explicit search for individual dimensions. With the techniques from the proofs in Section 6.1, a general formula for the lower bound on the number of degenerated quartets can be constructed. The result is Proposition 8.

Proposition 8. *Let t be the number of degenerated quartets and $A \subset \mathbb{F}_3^d$ a cap. Then*

$$t \geq \left\lceil \frac{3^{-d}|A|^4 - 4|A|^2 + 3|A|}{8} \right\rceil.$$

Proof. Let $a, b, c, d \in A$. Define

$$T = \sum_{y \in \mathbb{F}_3^d} \sum_{a,b,c,d \in A} \zeta^{(a+b-c-d)\cdot y}.$$

On one hand,

$$\begin{aligned}
\sum_{y \in \mathbb{F}_3^d} \sum_{a,b,c,d \in A} \zeta^{(a+b-c-d)\cdot y} &= \sum_{y \in \mathbb{F}_3^d} \sum_{a \in A} \zeta^{a\cdot y} \sum_{b \in A} \zeta^{b \cdot y} \sum_{c \in A} \zeta^{-c\cdot y} \sum_{d \in A} \zeta^{-d \cdot y} \\
&= \sum_{y \in \mathbb{F}_3^d} U_y U_y \overline{U_y U_y} \\
&= \sum_{y \in \mathbb{F}_3^d \setminus \{0\}} U_y^2 \overline{U_y}^2 + U_0 U_0 \overline{U_0 U_0} \\
&= ||\mathbf{u^2}||^2 + |A|^4.
\end{aligned}$$

$||\mathbf{u}^2||^2 \geq 0$. Hence,

$$T \geq |A|^4. \tag{7.1}$$

On the other hand, it follows in the same way as in the proof of Lemma 2 that $\sum_{y \in \mathbb{F}_3^d} \zeta^{(a+b-c-d)\cdot y} = 0$ whenever $a + b - c - d \neq 0$. Since $\sum_{y \in \mathbb{F}_3^d} \sum_{a,b,c,d \in A} \zeta^{(a+b-c-d)\cdot y} = \sum_{a,b,c,d \in A} \sum_{y \in \mathbb{F}_3^d} \zeta^{(a+b-c-d)\cdot y}$, this means $\sum_{y \in \mathbb{F}_3^d} \sum_{a,b,c,d \in A} \zeta^{(a+b-c-d)\cdot y} = 0$. Therefore, assume $a + b - c - d = 0$.

To fulfill the condition $a + b - c - d = 0$, there are four possible types of quartets: all equal, two pairs of equal points, one pair of equal points or all distinct.

If the four points are equal, i.e. $a = b = c = d$, then

$$\sum_{a=b=c=d \in A} \zeta^{(a+b-c-d)\cdot y} = \sum_{a \in A} 1 = |A|.$$

If there are two pairs of equal points, i.e. $a = c$ and $b = d$ or $a = d$ and $b = c$ (if $a = b$ and $c = d$, then $a = b = c = d$), then

$$\sum_{a=c\neq b=d,\in A} \zeta^{(a+b-c-d)\cdot y} = \sum_{a,b\neq a\in A} 1 = |A|(|A|-1).$$

Since the other possibilitie to pair up the points yields the same result, the total contribution of this case is $2|A|(|A|-1)$.

If there is only one pair of equal points, i.e. $a = b \neq c \neq d \neq a$ or $c = d \neq a \neq b \neq c$ ($a = c$ implies $b = d$ and $a = d$ implies $b = c$), then

$$\sum_{a=b\neq c\neq d\neq a\in A} \zeta^{(a+b-c-d)\cdot y} = \sum_{a,c\neq a\in A} 1 = |A|(|A|-1),$$

because d is determined by a, b and c. Again, because of the two possibilities to have one equal pair, the total contribution sum equals $2|A|(|A|-1)$.

The last case is when the four points are distinct, i.e. $|\{a, b, c, d\}| = 4$. In this case, the connection with t is made. Since the points are dinstinct and satisfy the condition $a+b-c-d = 0$, it follows that $(b-a)-(c-a)-(d-a) = (b-a)-(c-a)-(b-c) = 0$. This means that the three vectors $(b-a), (c-a)$ and $(d-a)$ are linearly dependent and hence, lie in a plane. As these vector determine the four points, they also lie in a plane. Therefore, the points are a degenerated quartet. The ordering of the points in a degenerated quartet does not matter. Hence, all degenerated quartets correspond to twenty-four ordered quartets.

Now, it will be shown that not all orderings of each degenerated quartets are counted in T. Assume x_1, x_2, x_3 and x_4 are a degenerated quartet. Let x_1 and x_2 form a set with y_1, and x_3 and x_4 with y_2. From $x_1 + x_2 + y_1 = 0$ and $x_3 + x_4 + y_2 = 0$ it follows that $x_1 + x_2 - x_3 - x_4 = 0$ if and only if $y_1 = y_2$. Hence, $x_1 + x_2 - x_3 - x_4 = 0$ if and only if the line through x_1 and x_2 and the line through x_3 and x_4 intersect. A degenerated quartet forms a maximal 2-cap. Since there is only one type of maximal 2-cap under affine transformations [7], it suffices to check for one case what part of the ordenings are counted.

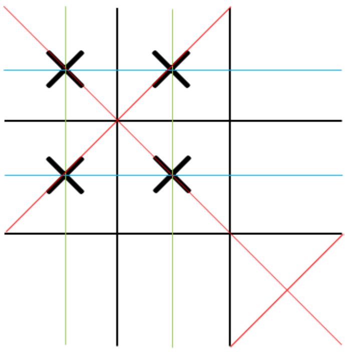

Figure 17: Possible pairs of lines through points of degenerated quartet.

Figure 17 shows that only one of the three combinations intersects. Therefore, only a third of the orderings of a degenerated quartet is counted. Concluding, the case that the four points are distinct, yields

$$\sum_{a,b,c,d,\in A} \zeta^{(a+b-c-d)\cdot y} = \frac{24t}{3} = 8t.$$

Adding up the obtained results for the different cases, yields

$$\sum_{a,b,c,d,\in A} \zeta^{(a+b-c-d)\cdot y} = |A| + 4|A|(|A|-1) + 8t.$$

Hence,

$$T = \sum_{y\in \mathbb{F}_3^d} \sum_{a,b,c,d,\in A} \zeta^{(a+b-c-d)\cdot y} = 3^d(|A| + 4|A|(|A|-1) + 8t). \tag{7.2}$$

Combining Equations 7.1 and 7.2, yields

$$3^d(|A| + 4|A|(|A|-1) + 8t) \geq |A|^4.$$

Together with the requirement on t being an integer this yields

$$t \geq \left\lceil \frac{3^{-d}|A|^4 - 4|A|^2 + 3|A|}{8} \right\rceil. \qquad \square$$

In Table 12 the lower bounds on t obtained by Proposition 8 are shown for dimensions four to ten. The lower bound does not only depend on the dimension, but also on the size of the (presumed) cap.

Dimension	4	4	5	5	5	5	6	6	6
Capsize	21	20	48	47	46	45	114	113	112
Lower bound	297	245	2725	2505	2295	2104	22505	21616	20751

Dimension	7	7	8	9	10
Capsize	291	290	771	2070	5619
Lower bound	372999	367627	6435277	114458743	2094463997

Table 12: Lower bounds on t by Proposition 8.

Now the results of Proposition 8 are established, they can be compared to the results from Section 5.4. In Section 5.4 there are two kinds of lower bounds on the number of degenerated quartets: one lower bound specifically for the maximal upper bound on the maximal size of a d-cap and one for the possible number of degenerated quartets based on the best known lower bounds.

In general, the more specific lower bounds from Section 5.4 are higher and the more general lower bouonds lower than the lower bounds from Proposiiton 8. Remarkably, for dimension four all possible number of degenerated quartets according to the result of the explicit search lie below the lower bound obtained from Proposition 5.4.

7.2 $\mathrm{U_y}$ and Hyperplane Triples

In this section the complex number U_y from the primitive root of unity method from Chapter 6 is associated with the hyperplane triples from the hyperplane counting method from Chapter 5. First, the data they represent is compared. This leads to the expressions from Proposition 9. In Proposition 10 the connection is strengthened with regard to the vector $\mathbf{u}$. Finally, it is shown that a solution of the ILP from Chapter 5 defines a vector $\mathbf{u}$.

Proposition 9. *The complex number U_y encodes the same data as the ordered hyperplane triple (ν_0, ν_1, ν_2) associated to y. In particular,*

$$U_y = \nu_0 + \nu_1\zeta + \nu_2\zeta^2$$

and

$$\begin{aligned}
\nu_0 &= \frac{2}{3}\operatorname{Re}(U_y) + \frac{1}{3}|A|,\\
\nu_1 &= \frac{1}{3}(|A| - \operatorname{Re}(U_y)) + \frac{1}{\sqrt{3}}\operatorname{Im}(U_y),\\
\nu_2 &= \frac{1}{3}(|A| - \operatorname{Re}(U_y)) - \frac{1}{\sqrt{3}}\operatorname{Im}(U_y).
\end{aligned}$$

Proof. It follows from the definitions of U_y and ν_c that

$$U_y = \sum_{a\in A}\zeta^{a\cdot y} = \sum_{c\in\mathbb{F}_3}\nu_c\zeta^c = \nu_0 + \nu_1\zeta + \nu_2\zeta^2.$$

Since

$$\overline{U_y} = \nu_0 + \nu_1\overline{\zeta} + \nu_2\overline{\zeta^2} = \nu_0 + \nu_1\zeta^2 + \nu_2\zeta,$$

$$U_y + \overline{U_y} = 2\nu_0 + (\nu_1+\nu_2)(\zeta+\zeta^2) = 2\nu_0 - \nu_1 - \nu_2 \tag{7.3}$$

and

$$U_y - \overline{U_y} = \nu_1(\zeta - \zeta^2) + \nu_2(\zeta^2 - \zeta) = (\nu_1 - \nu_2)(\zeta - \zeta^2). \tag{7.4}$$

Adding $\nu_0 + \nu_1 + \nu_2$ to Equation (7.3) yields

$$3\nu_0 = U_y + \overline{U_y} + \nu_0 + \nu_1 + \nu_2.$$

Since $U_y + \overline{U_y} = 2\operatorname{Re}(U_y)$ and $\nu_0 + \nu_1\nu_2 = |A|$,

$$\nu_0 = \frac{2}{3}\operatorname{Re}(U_y) + \frac{1}{3}|A|.$$

From $\nu_0 + \nu_1 + \nu_2 = |A|$ it follows that

$$\nu_1 + \nu_2 = |A| - \frac{2}{3}\operatorname{Re}(U_y) + \frac{1}{3}|A| = \frac{2}{3}|A| + \frac{2}{3}\operatorname{Re}(U_y). \tag{7.5}$$

Equation (7.4) yields

$$\nu_1 - \nu_2 = \frac{U_y - \overline{U_y}}{\zeta - \zeta^2}.$$

Since, $U_y - \overline{U_y} = 2i\operatorname{Im}(U_y)$ and $\zeta + \zeta^2 = \zeta + \overline{\zeta} = 2i\operatorname{Im}(\zeta)$,

$$\nu_1 - \nu_2 = \frac{2i\operatorname{Im}(U_y)}{2i\operatorname{Im}(\zeta)} = \frac{\operatorname{Im}(U_y)}{\frac{1}{2}\sqrt{3}} = \frac{2}{\sqrt{3}}\operatorname{Im}(U_y). \tag{7.6}$$

Adding Equations (7.5) and (7.6) yields

$$2\nu_0 = \frac{2}{3}|A| + \frac{2}{3}\operatorname{Re}(U_y) + \frac{2}{\sqrt{3}}\operatorname{Im}(U_y).$$

Hence,

$$\nu_0 = \frac{1}{3}(|A| + \operatorname{Re}(U_y)) + \frac{1}{\sqrt{3}} \operatorname{Im}(U_y).$$

Substracting Equation (7.6) from Equation (7.5) yields in the same way

$$\nu_2 = \frac{1}{3}(|A| + \operatorname{Re}(U_y)) - \frac{1}{\sqrt{3}} \operatorname{Im}(U_y). \qquad \square$$

Proposition 10. *All six possible orderings of a hyperplane triple (ν_0, ν_1, ν_2) yield the same value of $|U_y|$.*

Proof. By the definition of the norm of a complex number,

$$|U_y| = \sqrt{\operatorname{Re}(U_y)^2 + \operatorname{Im}(U_y)^2}. \tag{7.7}$$

An expression of $\operatorname{Re}(U_y)$ in terms of ν_0, ν_1 and ν_2 can be obtained from Proposition 9. Rewriting the expressions for ν_0, ν_1 and ν_2 yields

$$\operatorname{Re}(U_y) = \frac{3}{2}\nu_0 - \frac{1}{2}|A|, \tag{7.8}$$

$$\operatorname{Re}(U_y) = |A| + \sqrt{3}\operatorname{Im}(U_y) - 3\nu_1 \tag{7.9}$$

and

$$\operatorname{Re}(U_y) = |A| - \sqrt{3}\operatorname{Im}(U_y) - 3\nu_2. \tag{7.10}$$

Adding four times Equation (7.8) and one time Equation (7.9) to Equation (7.10) yields

$$6\operatorname{Re}(U_y) = 6\nu_0 - 3(\nu_1 + \nu_2).$$

Hence,

$$\operatorname{Re}(U_y) = \nu_0 - \frac{1}{2}(\nu_1 + \nu_2). \tag{7.11}$$

In the same way an expression for $\operatorname{Im}(U_y)$ can be obtained. Rewriting the expression yields

$$\operatorname{Im}(U_y) = \sqrt{3}\nu_1 - \frac{1}{\sqrt{3}}|A| + \frac{1}{\sqrt{3}}\operatorname{Re}(U_y) \tag{7.12}$$

and

$$\operatorname{Im}(U_y) = -\sqrt{3}\nu_2 + \frac{1}{\sqrt{3}}|A| - \frac{1}{\sqrt{3}}\operatorname{Re}(U_y). \tag{7.13}$$

Adding Equations (7.12) and (7.13) yields

$$2\operatorname{Im}(U_y) = \sqrt{3}(\nu_1 - \nu_2).$$

Hence,

$$\operatorname{Im}(U_y) = \frac{\sqrt{3}}{2}(\nu_1 + \nu_2). \tag{7.14}$$

Combining Equations (7.7), (7.11) and (7.14) yields

$$
\begin{aligned}
|U_y| &= \sqrt{(\nu_0 - \frac{1}{2}(\nu_1+\nu_2)^2 + (\frac{\sqrt{3}}{2}(\nu_1-\nu_2))^2} \\
&= \sqrt{\nu_0^2 - \nu_0(\nu_1-\nu_2) + \frac{1}{4}(\nu_1+\nu_2)^2 + \frac{3}{4}(\nu_1-\nu_2)^2} \\
&= \sqrt{\nu_0^2 - \nu_0(\nu_1-\nu_2) + \frac{1}{4}(\nu_1+\nu_2)^2 - \frac{1}{4}(\nu_1-\nu_2)^2 + (\nu_1-\nu_2)^2} \\
&= \sqrt{\nu_0^2 - \nu_0\nu_1 - \nu_0\nu_2 + \nu_1\nu_2 + (\nu_1-\nu_2)^2} \\
&= \sqrt{(\nu_0-\nu_1)(\nu_0-\nu_2) + (\nu_1-\nu_2)^2}.
\end{aligned}
$$

It can be concluded that $(\nu_0 - \nu_1)(\nu_0 - \nu_c) + (\nu_1 - \nu_2)^2 = \mathrm{Re}(Uy)^2 + \mathrm{Im}(U_y)^2$. With the equalities for ν_0, ν_1 and ν_2 from Proposition 9 both $(\nu_2 - \nu_1)(\nu_2 - \nu_0) + (\nu_1 - \nu_0)^2$ and $(\nu_1 - \nu_0)(\nu_1 - \nu_2) + (\nu_0 - \nu_2)^2$ can also be rewritten to $\mathrm{Re}(Uy)^2 + \mathrm{Im}(U_y)^2$. Hence,

$$
\begin{aligned}
|U_y| &= \sqrt{(\nu_0-\nu_1)(\nu_0-\nu_2) + (\nu_1-\nu_2)^2} \\
&= \sqrt{(\nu_1-\nu_2)(\nu_0-\nu_2) + (\nu_0-\nu_1)^2} \\
&= \sqrt{(\nu_1-\nu_0)(\nu_1-\nu_2) + (\nu_0-\nu_2)^2} \\
&.
\end{aligned}
$$

This means that the order in which ν_0, ν_1 and ν_2 appear in the hyperplane triple does not matter for the value of $|U_y|$. □

With the knowledge of Proposition 10 a link between the ILP from the hyperplane counting method and the vector **u** from the Fourier transform method can be established.

Remember that a solution of the ILP is a list of integers which represent the number of times a certain (unordered) hyperplane triple can occur. By the first constraint of the ILP, the integers add up to $\frac{3^d-1}{2}$.

On the other hand, remember that **u** is a vector of length $3^d - 1$ whose entries are $|U_y|$, paramatrized by $y \in \mathbb{F}_3^d \backslash \{0\}$. From Proposition 10 it follows that the entries in **u** are determined by the possible hyperplane triples. Hence, each hyperplane triple (ν_0, ν_1, ν_3) determines $x_{\nu_0\nu_1\nu_2}$ entries of **u**. Those entries will have the value of $|U_y|$ corresponding to the hyperplane triple (ν_0, ν_1, ν_3) according to the proof of Proposition 10. In this manner $\frac{3^d-1}{2}$ entries of **u** are determined. To obtain the other half of the entries it suffices to realise that with y and $-y$ the same case is considered. Hence, each integer in the solution of the ILP determines twice as many entries. Herewith, **u** is fully determined.

8 Conclusion

Since the maximal cardianality of a cap in $\mathbb{F}_3^d$ is only known for the dimensions one to six, the point of focus in this thesis was to establish upper bounds for the maximal cardinality of d-caps for the dimension seven to ten.

The two recursive methods that are described and applied follow from the articles by Davis and Maclagan [7] and Bierbrauer and Edel [3]. The first method, based on the counting of hyperplanes and hyperplane triples, is both simplified and extended. Therefore, there are three kinds of results obtained from this method. The second method is based on the Fourier transform. It leads to a direct expression for the upper bound on C_d, the maximal cardinality of a cap in $\mathbb{F}_3^d$. The results of the two methods from Chapters 5 and 6 are summarized in Table 13 below.

Dimension	1	2	3	4	5	6	7	8	9	10
Known value C_d	2	4	9	20	45	112	-	-	-	-
Lower bound	2	4	9	20	45	112	236	496	1064	2240
Upper bound - ILP	-	4	9	20	48	114	291	771	2070	5619
Upper bound - LP	-	4	9	21	48	114	292	771	2070	5619
Upper bound - 4-marked	-	-	-	20	48	114	291	771	2070	-
Upper bound - method 2	-	4	9	21	48	114	292	771	2070	5619
Best upper bound	-	4	9	20	45	112	291	771	2070	5619

Table 13: Lower and upper bounds on C_d by methods 1 and 2.

The different kinds of results from the first method are similar to each other. In the higher dimensions there is no difference at all. Therefore, the benefits of leaving the integer constraints out, which results in a faster program, outweigh the chance on a sharper bound.

The possible difference in results between the LP and the ILP can be retrieved by adding extra constraints based on the n-marked hyperplanes. This is done by adding the constraint based on 4-marked hyperplanes for the dimensions four to nine in Section 5.4. Even without determining the exact number of degenerated quartets, the obtained results were equivalent to the results from the ILP.

The results obtained with the second method using the best known upper bound on C_{d-1} are equal to the results from the LP of the first method. The second method has the preference since the results follow from an inequality only depending on C_d and the current dimension.

To conclude, the results on the upper bounds on the maximal cardinality of d-caps are compared with the lower bounds from Chapter 3. Since the exact value of C_d is known for dimension one to six, the best known upper and lower bounds both equal C_d. For the higher dimension, the difference between the lower and the upper bound grows rapidly. Whereas in dimension seven the difference is roughly a fifth of the upper bound, it is three-fifths in dimension ten.

9 Discussion

While there are little differences in the results obtained form the different methods described in this thesis, the approaches seem to lie further apart. For example, the running time of the corresponding programs or the required preparation.

The solution of the ILP contains only a small amount of nonzero values. This might suggest that the ILP can be simplified by reducing the number of variables without losing the accuracy. In the LP this is less relevant because there are more nonzero values in the solution.

The possible difference in results between the LP and the ILP can be retrieved by adding extra constraints based on the n-marked hyperplanes. In Section 5.4, a start has been made by adding the constraint based on 4-marked hyperplanes. Even without determining the exact number of degenerated quartets, the obtained results were equivalent to the results from the ILP.

When the exact number of degenerated quartets in a dimension is known, the bound could even be better. Adding a constraint based on 5-marked hyperplanes (and subsequently 6-marked and higher marked hyperplanes) may also lead to lower upper bounds. Similarly to the constraint for 4-marked hyperplanes, it would only depend on the number of quintets that span a three-dimensional space instead of a four-dimensional space since the maximal number of points from a cap in a plane is four.

At first sight, the two methods do not seem to use the same information because of the different approaches. However, the preparation of the inequality of the second method involves information and requirements similar to the assumptions and constraints in the hyperplane counting method.

In this thesis only a small step in linking the two methods together has been taken. Since the links between the hyperplane triples and U_y and the solution of the ILP and the vector $\mathbf{u}$ are only established to a certain extent, it would be interesting to develop them further. Moreover, a more general connection between the solutions of the (I)LP and the vector $\mathbf{u}$ including the constraints within the methods could be established.

Overall, it is clear that the upper bounds, as well as the lower bounds, should be a lot more improved before they can give a genuine idea of the value of C_d.

A Code Method 1

A.1 Python Code

```
from math import factorial

def choose(n,k):
    """Calculates the value of n choose k."""

    if k <= 0 or k > n:
        return 0

    return int(factorial(n)/(factorial(n-k)*factorial(k)))


def HT(capsize, prevmaxcap):
    """Lists all possiblie  hyperplane triples for sets of sze capsize based
    on the maximum capsize in the previous dimension (prevmaxcap)."""

    A = []
    p = prevmaxcap
    while p > (capsize -1)/3:
        q = prevmaxcap
        while q > (capsize - 4)/3:
            r = capsize - p - q
            z = [p,q,r]
            z.sort(reverse = True)
            if r >=0 and r <= q and z not in A:
                A.append(z)
            q -= 1
        p -= 1
    return A


def NH(dim, k):
    """Calculates the number of hyperplanes containing a k-dimensional subspace
    of F_3^dim."""

    return (3**(dim - k) - 1)/2


from choose import choose
from hyperplanetriples import HT, NH
from gurobipy import *

def capcheck(dim, capsize, prevmaxcap):
    """Checks with an (I)LP whether the set of size capsize has a nonnegative
    (integer) solution."""

    hts = HT(capsize, prevmaxcap)
    k = []
```

```
    l = []
    m = []
    r = 0
    s = 0
    u = 0
    for i in range(len(hts)):
        r = choose(hts[i][0],1) + choose(hts[i][1],1) + choose(hts[i][2],1)
        s = choose(hts[i][0],2) + choose(hts[i][1],2) + choose(hts[i][2],2)
        u = choose(hts[i][0],3) + choose(hts[i][1],3) + choose(hts[i][2],3)
        k.append(r)
        l.append(s)
        m.append(u)

    A = [k, l, m]
    b = [int(NH(dim, 1-1)*choose(capsize,1)), int(NH(dim, 2-1)*choose(capsize,2)),
         int(NH(dim, 3-1)*choose(capsize,3))]

    M = Model("capcheck")
    x = M.addVars(range(len(hts)), lb = 0, ub = GRB.INFINITY, vtype = GRB.INTEGER,
                  name = "x") #ILP
    x = M.addVars(range(len(hts)), lb = 0, ub = GRB.INFINITY, vtype = GRB.CONTINUOUS,
                  name = "x") #LP

    f = 0
    g = 0
    h = 0
    for i in range(len(hts)):
        f += k[i]*x[i]
        g += l[i]*x[i]
        h += m[i]*x[i]
    M.addConstr(f == b[0])
    M.addConstr(g == b[1])
    M.addConstr(h == b[2])

    M.optimize()

    if M.status != GRB.INFEASIBLE:
        return M.getAttr("x")

    else:
        return []


from capcheck import capcheck

def findmaxcap (dim, capsize, prevmaxcap):
    """Finds the highest number smaller than or equal to capsize that yields
    a nonnegative (integer) solution in capcheck."""

    while capsize > prevmaxcap:
```

```
        cc = capcheck(dim, capsize, prevmaxcap)
        if len(cc) == 0:
            capsize -= 1
        else:
            break

    return capsize
```

A.2 Matlab Code

```
function [res] = choose(n,k)
%CHOOSE Calculates the value of n choose k.

if k > n || k < 1
    res = 0;
else
    res = nchoosek (n,k);
end
end

function [res] = HT(capsize, prevmaxcap)
%HT Lists all possible hyperplane triples for sets
%of size capsize based on the maximum capsize in
%the previous dimension (prevmaxcap).

n = 0;
p = prevmaxcap;
while p > (capsize -1)/3
    q = prevmaxcap;
    while q > (capsize - 4)/3
        r = capsize - p - q;
        z = sort([p,q,r], 'descend');
        if r >= 0 && r <= q %&& in(A, z) == 0
            n = n+1;
        end
        q = q - 1;
    end
    p = p - 1;
end
res = {};
m = 1;
p = prevmaxcap;
while p > (capsize -1)/3
    q = prevmaxcap;
    while q > (capsize - 4)/3
        r = capsize - p - q;
        z = sort([p,q,r], 'descend');
        if r >= 0 && r <= q && ~(any(cellfun(@(x) isequal(x, z), res)))
            res{m} = z;
            m = m + 1;
```

```
        end
        q = q - 1;
    end
    p = p - 1;
end
end

function [res] = NH(dim, k)
%NH Calculates the number of hyperplanes containing
%a k-dimensional subspace of F_3^dim.

res = (3^(dim-k) -1)/2;
end

function [res] = capcheck(dim, capsize, prevmaxcap)
%CAPCHECK Checks with an (I)LP whether the set of
%size capsize has a nonnegative (integer) solution.

hts = HT(capsize, prevmaxcap);
v = size(hts);
lhts = v(2); %number of hyperplane triples
k = ones(1,lhts);
l = zeros(1,lhts);
m = zeros(1,lhts);
for i = 1:lhts
    s = 0;
    t = 0;
    for j = 1:3
        s = s + choose(hts{i}(j),2);
        t = t + choose(hts{i}(j),3);
    end
    l(i) = s;
    m(i) = t;
end
f = ones(1, lhts);
intcon = [1:lhts]; %ILP
intcon = []; %LP
A = [];
b = [];
Aeq = [k; l; m];
beq = [NH(dim, 1-1), NH(dim, 2-1)*choose(capsize,2), NH(dim, 3-1)*choose(capsize,3)];
lb = zeros(1, lhts);
ub = Inf(1, lhts);
res = intlinprog(f,intcon,A,b,Aeq, beq, lb, ub);
end

function [res] = findmaxcap(dim, capsize, prevmaxcap)
%FINDMAXCAP Finds the highest number smaller than or
%equal to capsize that yields a nonnegative
%(integer) solution in capcheck.
```

```
while capsize > prevmaxcap
    cc = capcheck(dim, capsize, prevmaxcap);
    v = size(cc);
    if v(1) == 0
        capsize = capsize - 1;
    else
        break
    end
end
res = capsize;
end
```

A.3 Code 4-Marked

```
from choose import choose
from hyperplanetriples import HT, NH
from gurobipy import *

def capcheck4m(dim, capsize, prevmaxcap, t):
    """... """

    hts = HT(capsize, prevmaxcap) #list of possible hyperplane triples
    k = []
    l = []
    m = []
    n = []
    r = 0
    s = 0
    u = 0
    v = 0
    for i in range(len(hts)):
        r = choose(hts[i][0],1) + choose(hts[i][1],1) + choose(hts[i][2],1)
        s = choose(hts[i][0],2) + choose(hts[i][1],2) + choose(hts[i][2],2)
        u = choose(hts[i][0],3) + choose(hts[i][1],3) + choose(hts[i][2],3)
        v = choose(hts[i][0],4) + choose(hts[i][1],4) + choose(hts[i][2],4)
        k.append(r)
        l.append(s)
        m.append(u)
        n.append(v)

    A = [k, l, m, n]

    b = [int(NH(dim, 1-1)*choose(capsize,1)), int(NH(dim, 2-1)*choose(capsize,2)),
         int(NH(dim, 3-1)*choose(capsize,3)), int(NH(dim, 4-1)*(choose(capsize,4)-t)
                                                 + NH(dim, 3-1)*t)]
    M = Model("capcheck4m")
    x = M.addVars(range(len(hts)), lb = 0, ub = GRB.INFINITY, vtype = GRB.CONTINUOUS,
                  name = "x")
```

```
    f = 0
    g = 0
    h = 0
    j = 0
    for i in range(len(hts)):
        f += k[i]*x[i]
        g += l[i]*x[i]
        h += m[i]*x[i]
        j += n[i]*x[i]
    M.addConstr(f == b[0])
    M.addConstr(g == b[1])
    M.addConstr(h == b[2])
    M.addConstr(j == b[3])

    M.optimize()

    if M.status != GRB.INFEASIBLE:
        return M.getAttr("x")

    else:
        return []

from __future__ import division
from capcheck4marked import capcheck4m
from choose import choose
import math

def findmaxcap4m (dim, capsize, prevmaxcap, t):
    """..."""
    while capsize > prevmaxcap:
        cc = capcheck4m(dim, capsize, prevmaxcap, t)
        if len(cc) == 0:
            capsize -= 1
        else:
            break

    return capsize


import matplotlib.pyplot as plt

t1 = [0, 500, 1500, 5000, 7579, 7580, 10000, 20000, 24497, 35000, 44000,
      46000, 50000, 96000, 150000, 185000, 200000, 248497, 350000, 400000,
      410000, 420000, 430000, 440000, 447107]
t2 = [447108, 447150, 448000, 448492]
t3 = [448493, 451260]
t4 = [451261, 500000, 513372, 610000, 690029, 770058, 898401, 1026745]
y1 = [112, 112, 112, 112, 112, 113, 120, 141, 153, 161, 170, 171, 174,
      203, 225, 236, 240, 253, 274, 283, 284, 286, 288, 289, 290]
```

```
y2 = [112, 112, 112, 112]
y3 = [291, 291]
y4 = [112, 112, 112, 112, 112, 112, 112, 112]

a = [0, 1026745]
b = [236, 236]
c = [112, 112]

plt.xlabel('number of degenerated quartets (t)')
plt.ylabel('upper bound on maximal size 7-cap (y)')

plt.plot(a, c, 'y', linestyle = '--', label = 'y = 112')
plt.plot(a, b, 'g', linestyle = '--', label = 'y = 236')

plt.axvline(185000, color = 'm', linestyle = '--', label = 't = 185000') #full plot
plt.axvline(451260, color = 'r', linestyle = '--', label = 't = 451260') #full plot
plt.axvline(447107, ymin = .06, ymax = 0.95, color = 'b', linestyle = '--',
            label = 't = 44707') #zoom plot
plt.axvline(448493, ymin = .06, ymax = 0.955, color = 'b', linestyle = '--',
            label = 't = 448493') #zoom plot
plt.axvline(451260, ymin = .06, ymax = 0.955, color = 'b', linestyle = '--',
            label = 't = 451260') #zoom plot
plt.axvline(447107, ymin = .06, ymax = 0.95, color = 'b', linestyle = '--') #full plot
plt.axvline(448493, ymin = .06, ymax = 0.955, color = 'b', linestyle = '--')#full plot
plt.axvline(451260, ymin = .06, ymax = 0.955, color = 'b', linestyle = '--')#full plot

plt.plot(t1,y1,'b', t2, y2, 'b', t3, y3, 'b', t4, y4, 'b')

plt.axis([0, 1026745, 100, 300]) #full plot
plt.axis([444000, 454000, 100, 300]) #zoom plot

plt.legend(fontsize = 11.5)

plt.show()
```

A.4 Figures 4-Marked

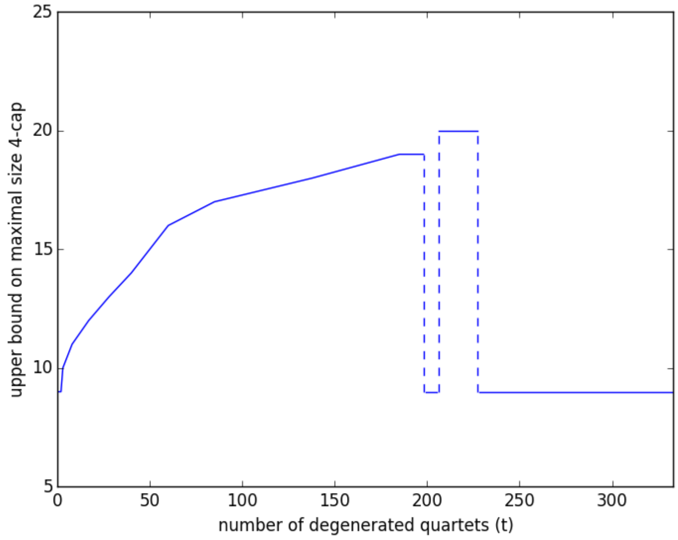

Figure 18: Upper bounds on C_4 for varying t.

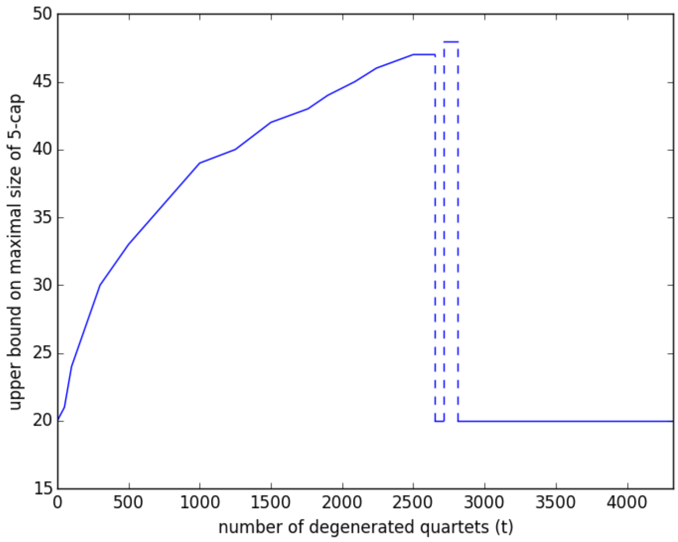

Figure 19: Upper bounds on C_5 for varying t.

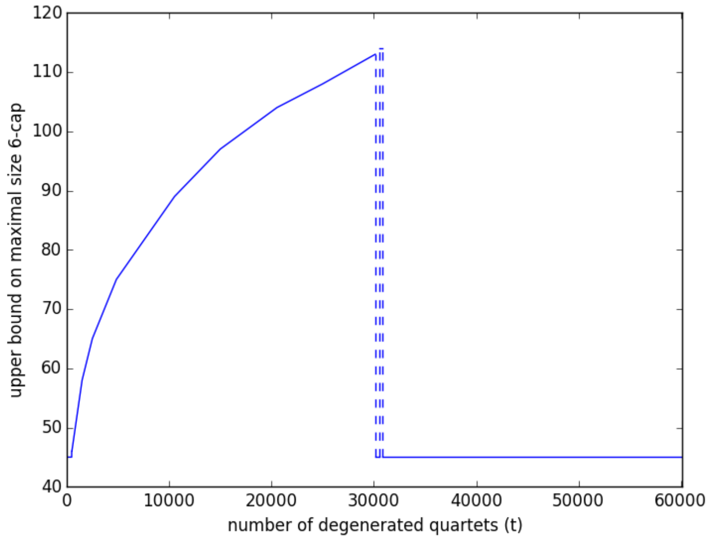

Figure 20: Upper bounds on C_6 for varying t.

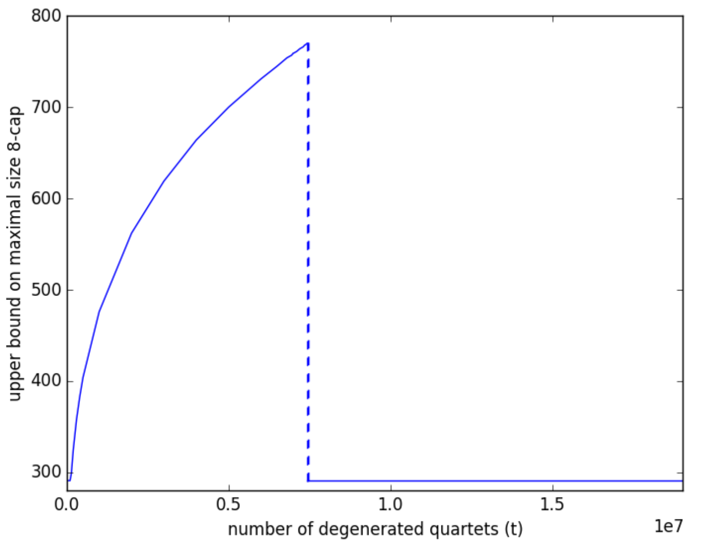

Figure 21: Upper bounds on C_8 for varying t.

B Code Method 2

```
from __future__ import division
import math

def corollary2_1(dim, prevmaxcap):
    """Obtains an upper bound for C_d by Corollary 2.1."""

    ub = math.floor((1 + 3*prevmaxcap)/(1 + 3**(-dim +1)*prevmaxcap))
    print int(ub)
```

References

[1] M. Berger. *Geometry I.* Springer Science & Business Media, 1987.

[2] J. Bierbrauer. *Introduction to coding theory.* CRC Press, 2016.

[3] J. Bierbrauer and Y. Edel. Bounds on affine caps. *Journal of Combinatorial Designs*, 10(2):111–115, 2002.

[4] A. E. Brouwer, J. B. Shearer, N. J. A. Sloane, and W. D. Smith. A new table of constant weight codes. *IEEE Transactions on Information Theory*, 36(6):1334–1380, 1990.

[5] A. R. Calderbank and P. C. Fishburn. Maximal three-independent subsets of $\{0, 1, 2\}^n$. *Designs, Codes and Cryptography*, 4(4):203–211, 1994.

[6] P. J. Cameron. *Projective and polar spaces*, volume 13. University of London, Queen Mary and Westfield College, 1992.

[7] B. L. Davis and D. Maclagan. The card game SET. *The Mathematical Intelligencer*, 25(3):33–40, 2003.

[8] Y. Edel. Extensions of generalized product caps. *Designs, Codes and Cryptography*, 31(1):5–14, 2004.

[9] Y. Edel and J. Bierbrauer. Large caps in small spaces. *Designs, Codes and Cryptography*, 23(2):197–212, 2001.

[10] L. Ji. Asymptotic determination of the last packing number of quadruples. *Designs, Codes and Cryptography*, 38(1):83–95, 2006.

[11] A. C. Mukhopadhyay. Lower bounds on $m_t(r, s)$. *Journal of Combinatorial Theory, Series A*, 25(1):1–13, 1978.

[12] G. Pellegrino. Sul massimo ordine delle calotte in S4, 3. [The maximal order of the spherical cap in S4,3]. *Le Matematiche*, 25(10):149–157, 1970.

[13] SET Enterprises. Founder & Inventor: Marsha J. Falco. `http://www.setgame.com/founder-inventor`. Acessed: 17 May 2017.

[14] E. Snapper and R. J. Troyer. *Metric affine geometry.* Academic Press, 1971.