2011

Algebraic Codes For Error Correction In Digital Communication Systems

Jibril, Mubarak

Copyright Statement

Algebraic Codes For Error Correction In Digital Communication Systems

by

Mubarak Jibril

A thesis submitted to the University of Plymouth
in partial fulfilment for the degree of

Doctor of Philosophy

School of Computing and Mathematics
Faculty of Technology

Under the supervision of Dr. M. Z. Ahmed,
Prof. M. Tomlinson and Dr. C. Tjhai

December 2011

ABSTRACT

Algebraic Codes For Error Correction In Digital Communication Systems

Mubarak Jibril

C. Shannon presented theoretical conditions under which communication was possible error-free in the presence of noise. Subsequently the notion of using error correcting codes to mitigate the effects of noise in digital transmission was introduced by R. Hamming. Algebraic codes, codes described using powerful tools from algebra took to the fore early on in the search for good error correcting codes. Many classes of algebraic codes now exist and are known to have the best properties of any known classes of codes. An error correcting code can be described by three of its most important properties length, dimension and minimum distance. Given codes with the same length and dimension, one with the largest minimum distance will provide better error correction. As a result the research focuses on finding improved codes with better minimum distances than any known codes.

Algebraic geometry codes are obtained from curves. They are a culmination of years of research into algebraic codes and generalise most known algebraic codes. Additionally they have exceptional distance properties as their lengths become arbitrarily large. Algebraic geometry codes are studied in great detail with special attention given to their construction and decoding. The practical performance of these codes is evaluated and compared with previously known codes in different communication channels. Furthermore many new codes that have better minimum distance to the best known codes with the same length and dimension are presented from a generalised construction of algebraic geometry codes. Goppa codes are also an important class of algebraic codes. A construction of binary extended Goppa codes is generalised to codes with nonbinary alphabets and as a result many new codes are found. This construction is shown as an efficient way to extend another well known class of algebraic codes, BCH codes. A generic method of shortening codes whilst increasing the minimum distance is generalised. An analysis of this method reveals a close relationship with methods of extending codes. Some new codes from Goppa codes are found by exploiting this relationship. Finally an extension method for BCH codes is presented and this method is shown be as good as a well known method of extension in certain cases.

CONTENTS

List of Figures

List of Tables

List of Algorithms

ACKNOWLEDGEMENTS

My eternal gratitude goes to my parents Prof. and Mrs Jibril for their love and support over the years.
I would like to thank my supervisors Prof Martin Tomlinson, Dr Ahmed Zaki and Dr Cen Tjhai for their invaluable guidance and support during the course of my PhD.
I would also like to thank my brothers and sisters Amal, Muhsin, Ihsan, Halima and Khidir for their kindness and support.
I would also like to thank my fiance, Mubaraka for her love and support during the years.
I would like to acknowledge scholarships and awards I have received from the Petroleum Technology Trust Fund (PTDF) in Nigeria and the Dean's Scholarship from the Faculty of Technology.
I would like to thank Prof. Emmanuel Ifeachor for his advice during the course of my PhD.
I am grateful to friends and colleagues Li Yang, Jing Cai, Ismail Isnin, Purav Shah, Athanasios Anastasiou, Peng Zhao, Nadia Awad and Cemil Kilerci for all the good times and for being good company.

AUTHOR'S DECLARATION

At no time during the registration for the degree of Doctor of Philosophy has the author been registered for any other University award without prior agreement of the Graduate Committee. This study was partly financed by the Dean's Scholarship of the Faculty of Technology, University of Plymouth and by the Petroleum Technology Development Trust Fund (PTDF), Nigeria. Results culminating from research have been regularly published in peer reviewed conferences.

Publications:

'Performance comparison between Hermitian and Nonbinary BCH Codes'. *International IEEE conference on microwaves, communications, anntennas and electronic systems (COMCAS), Proceedings on.*, Nov, 2009.
'Good codes from generalised algebraic geometry codes'. *IEEE symposium on information theory (ISIT), Proceedings on.*, July, 2010.
'Improvements on codes from nonbinary fields using generalised algebraic geometry codes'. *IEEE international conference on wireless communication, networking and information security (WCNIS), Proceedings on.*, August, 2010.
'New binary codes from extended Goppa codes'.*Accepted to the third international castle meeting on coding theory and applications (ICMCTA)*, September, 2011. Thesis

main body word count: 23350 words

Signed ..... Date .....

Part I

Introduction, Definitions and Preliminaries

1. Introduction and Motivation

The field of coding theory was born with Claude E. Shannon's (Shannon, 1948) landmark work on digital communications. Digital communications involves the transmission of message signals in digital form. Shannon's work shed light on the limits of how fast, efficient and reliable a digital communication system can be through the most significant transmission channels. Shannon's work specifically focuses on how to transmit data reliably through channels where signal altering noise is present. A key observation was that it was possible to increase the reliability of a digital communication system significantly by using efficient error correction schemes. Digital signals to be transmitted are encoded with a predefined *code* at the transmitting end and then suitably decoded at the receiving end.

Conceptually, a digital communication system with an error correction scheme is modelled as in Figure 1.1. The source encoder (not shown) accepts source data usually in binary form and compresses it by exploiting its inherent redundancy. The compression is done in such a manner that the source decoder can recover the information without any ambiguity. Compressed data from the source encoder is fed to the channel encoder at the rate of R bits per second. A channel encoder adds redundancy to k bits [1] to produce n bits where $n \geq k$. The channel encoder block outputs data at a rate of $R = \frac{n}{k}$ bits per second. This block essentially begins the error correction scheme. It uses error correction codes which can either be linear block codes or convolutional codes. The next block is the modulator whose purpose is to transform the signal so that it is suitable for transmission through the channel. A chosen modulation scheme will take into consideration the specific channel characteristics. The output of the modulator is sent through the channel where it is distorted by noise. A channel can be defined by the type of distortion it contributes to the transmitted signal. Additive white Gaussian noise channels are channels where an additive, zero mean and Gaussian distributed noise dominates while fading channels have interference noise due to the fact that signals travel in multiple paths. From a coding scheme design perspective, a channel can also be defined by its limitations on power or bandwidth. The demodulator attempts to reproduce the transmitted signal from the modulator from the impaired received signal. It achieves this by using a threshold value on a per bit basis. If a single threshold is used, then the output of the demodulator is called hard and the demodulator is said

[1] *In general the channel encoder need not be a binary encoder and can process non-binary symbols.*

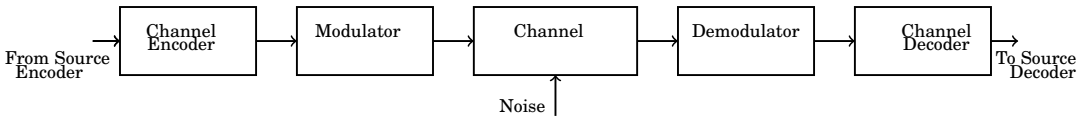

Fig. 1.1: Digital Communication System

to perform hard decisions. If more than one threshold value is used, the demodulator produces soft outputs and performs soft decisions. The type of decision taken by the demodulator decides the kind of channel decoder used in the channel decoder block. Soft/hard decision channel decoders input soft/hard decided sequences from the demodulator and using decoding algorithms attempt to detect and correct errors incurred by transmission. Finally a source decoder decompresses the output of the channel using a decompression algorithm to produce an estimate of the original message. If the channel decoder succeeds in correcting all errors then the output of the source decoder is the original message.

The scope of this research lies between the channel encoder and the channel decoder. The research focuses mainly on finding good linear error correcting codes that improve the reliability of digital communications. Specifically, the thesis focuses on an important sub-class of linear codes, algebraic geometry codes, and codes they generalise. As a welcomed necessity, the research navigates through the abundant literature of algebraic codes and their equally rich algebraic structures.

1.1 Shannon's Contributions and Implications

Shannon's main result is the relationship (Shannon, 1948),

$$R < W \log_2(1 + \text{SNR}) \tag{1.1}$$

for the additive white Gaussian noise (AWGN), where R is transmission data rate in bits per second, W is the transmission channel bandwidth in hertz (Hz) and SNR is the average signal power to the average noise power ratio, a dimensionless quantity. Equation (1.1) is commonly referred in literature as the Shannon or Shannon-Hartley equation. The inequality may be removed and R replaced with the "channel capacity", C, and obtain a limit on the rate so that

$$R < C = W \log_2(1 + \text{SNR}). \tag{1.2}$$

In order to achieve capacity rates it can see from (1.2) that it is possible to increase either the bandwidth W or the signal to noise ratio SNR or both. Since channel noise cannot be controlled by the system designer, increasing SNR is only possible by increasing the signal power since

$$\text{SNR} = \frac{S}{N} \tag{1.3}$$

where S and N represent the average signal and noise power respectively. Prior to channel coding and Shannon's postulations increasing transmission power was synonymous with increased reliability. There are however limits to how much power is available for specific applications. Increasing the bandwidth W might seem like a good solution until one considers that

(i) channel bandwidth is limited,

(i) the additional cost and size of transmission equipment.

In addition, for the AWGN channel the average signal noise $N = N_0 W$ where N_0 is the noise spectral density. So as the bandwidth increases, the SNR deteriorates so does the rate albeit logarithmically. In summary Shannon was able to show that noise in channel limits the rate of data transmission through the channel but does not limit the reliability and accuracy of transmission. In so far as the data transmission rate is below channel capacity, reliable transmission is possible.

1.2 History and Development of Coding Theory

1.2.1 1948–1970

An important and perhaps understated contribution to the field of coding theory was due to Hamming (1950). Whereas Shannon's approach was non-constructive and asymptotic, Hamming's contribution was constructive and combinatorial (Sudan, 2001). Shannon showed that error free communication at rates below capacity is possible if *good* error correcting codes are used. However it soon became clear that finding error correcting codes with efficient decoding schemes that fit this description was difficult. Hamming was the first to present non-trivial error correcting codes. These codes are known aptly as Hamming codes. A code is traditionally defined with there parameters (i) its length or number of symbols (ii) its dimension or number of message symbols (iii) and its minimum distance or the smallest possible distance between any two distinct code words. The problem of designing good codes preoccupied coding theorists in the aftermath of these contributions. Golay (1949) presented codes known today as Golay codes. These codes include a generalisation of Hamming codes and also some *perfect* codes. The word perfect used here is in retrospect a misnomer, these codes were not perfect in the sense that they could correct all forms of error but they were labelled perfect because of their unique structural properties. Perfect codes have the property that every corrupted codeword can be uniquely decoded without ambiguity. Reed (1954) and Muller (1954) presented an important class of codes called Reed-Muller codes. Hamming, Golay and Reed-Muller codes have interesting algebraic properties and continue to be the subject of theoretical investigations. Elias (1955) invented convolutional codes. Convolutional codes were not described algebraically (although they can be) and are

the first implementation of probabilistic codes. Convolutional codes were soon found to be equal or better than block codes for almost any practical application (Forney, 1970). Binary Bose Chaudhari Hocquenghem (BCH) were discovered by Bose and Chaudhuri (1960) and independently by Hocquenhem (1959). Binary BCH codes were the first powerful error correcting codes with a potential for practical use. BCH codes are known to contain Hamming codes as a subset. Reed Solomon (RS) codes were discovered by Reed and Solomon (1960). RS codes are a subset of BCH codes and have good properties. Despite being primarily non-binary codes, RS codes have found extensive practical use to this day. A year later Gorenstein and Zierler (1961) presented BCH codes with binary symbols which are a generalisation of RS codes. Though algebraic codes were presented with elegant theories and had exceptional properties it became difficult to find efficient and simple decoding schemes that took advantage of their error correcting capabilities. Peterson (1960) was the first to present a decoding algorithm for BCH (and by definition RS) codes that corrected errors up to the designed error correction capabilities of these codes. Soon afterwards Berlekamp (1968) presented a much simpler scheme than Peterson's algorithm. Massey (1969) showed that Berlekamp's algorithm solved a well known linear feedback shift register synthesis problem. The algorithm consequently became known as the Berlekamp Massey decoding algorithm. In the first 20 years since the birth of coding theory, powerful algebraic and probabilistic codes had already been invented with efficient decoding algorithms. BCH codes were however shown to be asymptotically bad[2] i.e. their error correction capabilities deteriorate as lengths approach infinity. Forney and Costello (2007) showed a method of concatenating RS codes with short codes and proved that these codes were asymptotically good. Concatenated codes proved useful in bursty channels where errors occur in short consecutive bursts in the transmitted data stream instead of independently.

1.2.2 1970-1990

Justesen (1972) produced the first class of constructive algebraic codes are asymptotically good. However Justesen's codes have found little use, since for practical lengths far superior codes exist. Goppa (1970) introduced a class of linear algebraic error correcting codes. These codes were a large class of codes and in some instances can be seen as generalisations of BCH codes. Patterson (1975) presented a decoding algorithm for Goppa codes obtained by modifying the Berlekamp Massey algorithm for BCH codes. Goppa codes are known to be asymptotically good (MacWilliams and Sloane, 1983). Goppa (1988) then presented a new class of algebraic codes from curves called algebraic geometry (AG) codes. These codes are generalisations of RS

[2] A code is said to be asymptotically good if its minimum distance increases proportionately as the length becomes arbitrarily large. This means that an extremely long code will have a good minimum distance.

and Goppa codes (of which BCH codes are subsets). Certain families of these codes were shown by Tsfasman et al. (1982) to be asymptotically better than previous algebraic codes. There was great interest in the decoding of AG codes up to their designed error correction capabilities. By the fourth decade from the birth of coding theory, efficient algebraic codes had been realised some of which have been shown to be asymptotically good. The major obstacle to achieving error-free communication for these codes was their decoding complexity when utilising channel statistics. All previously mentioned methods of decoding algebraic codes are bounded distance decoding schemes that do not exploit channel statistics in decoding but rely solely on the algebraic structure of the codes. It was clear that to achieve error-free communication one must utilise channel statistics. Although research concentrated on searching for codes with a good algebraic structure and predetermined properties it became increasingly apparent that to achieve error free communication random-like codes with acceptable properties and moderate decoding complexity were required. This line of thought was first pursued by Elias (1955) and subsequently by Gallager (1962).

1.2.3 1990–

Berrou et al. (1993) astounded the coding theory community by presenting Turbo codes which approached Shannon's limit closely. Turbo codes are derived from convolutional codes and are probabilistic codes. Research in the 1990's was then shifted from algebraic codes to probabilistic codes. MacKay and Neal (1996) soon reintroduced codes by Gallager (1962) showing that these code (called low density parity check (LDPC) codes) also approached Shannon's limit. By sacrificing exceptional properties for reduced decoding complexity coding theorists were able to achieve these results. Algebraic coding also received a significant boost in the introduction of bounded distance decoding for AG codes by Feng and Rao (1993) and Sakata et al. (1995). The decoding algorithm by Feng and Rao (1993) was a generalisation of the algorithm by Peterson and Weldon (1972) while the algorithm by Sakata et al. (1995) was a generalisation of the algorithm by Berlekamp (1968) and Massey (1969). An important milestone was reached for algebraic codes when Guruswami and Sudan (1999) presented a new algorithm for decoding BCH and RS codes that had better performance than previously known decoding techniques. Sudan's decoding used the notion of a list decoder that returned a list of candidates for each received sequence. Koetter and Vardy (2003) extended this algorithm to utilise channel statistics so as to further improve decoding performance. Sudan's algorithm was soon after extended to decoding AG codes by Shokrollahi and Wasserman (1999). A generalised construction of AG codes was presented by Xing et al. (1999a) which have better properties than AG codes.

Near Shannon limit performance is now obtainable from probabilistic codes like

LDPC and turbo codes. At present LDPC codes have shown the most promise in attaining error-free communication and current research trends are focused on constructing good codes. Some algebraic codes have the best distance properties but their decoding complexity using channel statistics is proving to be a stumbling block towards achieving error-free communication. However the search for improved algebraic codes with less complex and efficient decoders is still active since it is known that performance ultimately depends on the distance properties of a code.

1.3 Research Scope

The work presented in this thesis focuses mainly on algebraic codes. In particular AG codes and Goppa codes are the major subject of interest while related codes BCH and RS codes also feature prominently. AG codes were invented by Goppa (1988) but have not received as much research attention as other error correction codes. This is in part due to the fact that much of the theory of the codes is obtained from deep mathematical aspects of algebraic geometry that do not lend themselves to easy access. Subsequent advances have however simplified construction and decoding of the most popular AG codes. The thesis has two main objectives;

- To study the AG codes and related algebraic codes. The performance of AG codes relative to similar codes is investigated in different communication channels using different decoding approaches. The theory and properties of these codes are also studied.

- To obtain codes from algebraic codes with better minimum distances than any previously known codes.

1.4 Major Contributions of the Thesis

- Algebraic geometry codes are studied in great detail with special attention given to their construction and decoding. The practical performance of these codes is evaluated and compared with previously known codes in different communication channels. Decoding performance of AG and nonbinary BCH codes is compared in the AWGN using soft and hard decision decoding and erasure channels using maximum likelihood decoding. The Berlekamp Massey Sakata algorithm (BMSA) decoding is presented for the bounded distance decoding of AG codes while the classic Berlekamp Massey algorithm (BMA) is presented for BCH codes for transmission in the AWGN channel. Symbol based ordered reliability decoding is carried out for soft decision decoding in the AWGN channel for both AG and BCH codes. Finally maximum-likelihood erasure decoding (in-place) is presented for decoding these codes in the erasure channel.

- New codes that have better minimum distances than the best known codes with the same length and dimension are presented from a generalised construction of algebraic geometry codes. Using this method 237 codes in the finite field $\mathbb{F}_{16}$ from four curves with better minimum distances than any known codes are presented. Many improvements on constructible codes were also presented. Furthermore by applying simple modifications to the presented codes more improvements are possible.

- A construction of extended binary Goppa codes is generalised to codes with nonbinary alphabets and as a result new codes are found. This construction is shown to be an efficient way to extend another well known class of algebraic codes, BCH codes. In total 48 new codes in finite fields $\mathbb{F}_7$, $\mathbb{F}_8$ and $\mathbb{F}_9$ were presented directly from this method. With further extensions using construction X (MacWilliams and Sloane, 1983), 30 more improvements are also obtained. More improvements are also possible from simple modifications of the obtained codes.

- A generic method of shortening codes whilst increasing their minimum distances is generalised. An analysis of this method reveals a close relationship with methods of extending codes. Codes with a special structure from Goppa codes are used and this relationship is exploited to obtain 4 new binary codes.

- Finally an extension method for BCH codes is presented and this method is shown be as good as a well known method of code extension in certain cases.

1.5 Publication List

Published

M. Jibril, M. Tomlinson, M. Z. Ahmed and C. Tjhai. 'Performance comparison between Hermitian and Nonbinary BCH Codes'. *International IEEE conference on microwaves, communications, antennas and electronic systems (COMCAS), Proceedings on.*, Nov, 2009. http://dx.doi.org/10.1109/COMCAS.2009.5386010

M. Jibril, M. Tomlinson, M. Z. Ahmed and C. Tjhai. 'Good codes from generalised algebraic geometry codes'. *IEEE symposium on information theory (ISIT), Proceedings on.*, July, 2010.http://dx.doi.org/10.1109/ISIT.2010.5513687

M. Jibril, M. Tomlinson, M. Z. Ahmed and C. Tjhai. 'Improvements on codes from nonbinary fields using generalised algebraic geometry codes'. *IEEE international conference on wireless communication, networking and information security (WCNIS), Proceedings on.*, August, 2010.http://dx.doi.org/10.1109/WCINS.2010.5541920

M. Tomlinson, M. Jibril, C. Tjhai, M. Grassl and M. Z. Ahmed. 'New binary codes from extended Goppa codes'. *Accepted to the third international castle meeting on coding theory and applications (ICMTA)*, September, 2011.

Submitted

M. Tomlinson, M. Jibril, C. Tjhai, S. Bezzateev, M. Grassl and M. Z. Ahmed. 'A generalised construction and improvements on nonbinary codes from Goppa codes'. *To be submitted to the IEEE Transactions on Information Theory.*, July, 2011.

M. Jibril, S. Bezzateev, M. Tomlinson, C. Tjhai, M. Z. Ahmed. 'Some results from binary Goppa codes and a case of shortening linear codes'. *To be submitted to the IET Journal on Communications.*, July, 2011.

1.6 Thesis Organisation

- **Part I: Introduction and Motivation**
  - **Linear Codes Over Finite Fields**
    In this Chapter the concept of finite fields is introduced. Linear codes and their basic properties are then defined. Since the thesis focuses on constructing new codes, details of some well known generic constructions are also given. The most important type of channel models are also presented. This Chapter provides sufficient preliminary information relevant to subsequent Chapters.
- **Part II :Algebraic Codes for Error Correction**
  - **One Dimensional Codes:RS, BCH and Goppa codes**
    In this Chapter RS, BCH and Goppa codes are introduced. This Chapter serves as a precursor to subsequent Chapters in the thesis by introducing three important classes of codes.
  - **Two Dimensional Codes: AG Codes**
    AG codes are introduced in this Chapter. Their underlying theory and definition are then presented. Examples of constructions of AG codes are also given.
  - **Decoding Algebraic Codes**
    The decoding of algebraic codes for the error and erasure channels is discussed in this Chapter. The BMSA decoding is presented for the bounded distance decoding of AG codes while the classic BMA is presented for BCH codes for transmission in the AWGN channel. Ordered reliability decoding is presented for soft decision decoding in the AWGN channel.

Finally maximum-likelihood erasure decoding (in-place) is presented for decoding in the erasure channel.

  - **Performance of Algebraic Codes**
    Performance of AG codes is compared with shortened nonbinary BCH codes in the same finite field having the similar rate and length. The codes are compared the AWGN channel (soft and bounded decoding) and in the erasure channel. Conclusions are drawn from the results.

- **Part III: Search For New Codes**

  - **Introduction**
    This Chapter details the methods and approaches used in the search for new codes and introduces Part III of the thesis.
  - **Improved Codes From Generalised AG codes**
    This Chapter presents the concept of places of a curve of degree larger than one and generalised constructions of AG codes. As a result 237 new codes in the finite field $\mathbb{F}_{16}$ from three curves using a generalised construction of AG codes are presented.
  - **Improved Codes From Goppa Codes**
    This Chapter presents 108 improvements to the best known codes in finite fields $\mathbb{F}_7$, $\mathbb{F}_8$, $\mathbb{F}_9$ from extended Goppa codes. The method used is a generalisation of a well known method for extending binary Goppa codes to nonbinary finite fields.
  - **A Special Case Of Shortening Linear Codes**
    Theory and proof of a method of shortening linear codes is provided. The link between shortening and extending linear codes is then discussed. Four new binary codes obtained by exploiting this link from a Goppa code with a special structure.

- **Part Four: More On Algebraic Codes**

  - **Notes on Extending BCH Codes**
    A method of extending BCH codes is explored. This method is shown to be as good as the best generic method of extending codes in certain cases. The method provides insight into the limits of extendability of BCH codes.
  - **Improved Codes From Goppa Codes II**
    An alternative construction is presented in this chapter for extended Goppa codes. This construction produces shorter codes than the previously described method however provides greater flexibility in constructing codes.

2. Linear Codes Over Finite Fields

This chapter introduces the theory of finite or Galois fields. This theory forms the foundation upon which linear codes can be studied. Sufficient information is given on the finite fields section since it serves as a prerequisite for subsequent chapters. For an in depth treatment of the subject see Shu and Costello (2004), MacWilliams and Sloane (1983) and Lidl and Niederreiter (1986). The information provided on finite fields is obtained from these aforementioned sources. The chapter also introduces linear codes over finite fields. Both these topics are large and only information relevant to this thesis is provided.

2.1 Finite Fields

In mathematics, fields are loosely defined as algebraic structures that contain a set of elements for which the operations multiplication and addition (and their respective inverse operations division and subtraction) are clearly defined. In addition the result of any of these defined operations results in elements within the field. A typical example is the field of real numbers.

2.1 *Definition (Finite Field).* *A finite field denoted by $\mathbb{F}_{p^r}$ is a field with a finite number of elements where p is always prime and $r \geq 1$.*

p is called the *characteristic* of the finite field.

Field Elements: A finite field contains the basic elements of a field; a multiplicative identity element denoted by 1 and an additive identity element denoted by 0. The order of a finite field element α is the smallest integer n such that $\alpha^n = 1$. A primitive element of a finite field is any nonzero element α with order p^r.

2.2 *Definition (Subfield).* *A subfield $\mathbb{F}_{p^s}$ is a subset of the finite field $\mathbb{F}_{p^r}$ and contains*

only nonzero elements that satisfy,

$$\beta^{p^s-1} = 1 \qquad \beta \in \mathbb{F}_{p^r}$$

and $s|r$.

All finite fields $\mathbb{F}_{p^r}$ for which $r \geq 2$ are called extension fields of $\mathbb{F}_p$. Henceforth finite fields will be denoted by $\mathbb{F}_q$ where $q = p^r$ for brevity. Consider the univariate polynomial ring with coefficients in the finite field $\mathbb{F}_q$ denoted as $\mathbb{F}_q[x]$. An irreducible polynomial is a polynomial in this ring is one which is prime i.e. it cannot be factorised. A *primitive* polynomial of degree m is an irreducible polynomial which has the primitive element of $\mathbb{F}_{q^m}$ as a root. Using modulo operations, a primitive polynomial of degree m can be used to generate $\mathbb{F}_{q^m}$.

Example 2.1 (Finite Field $\mathbb{F}_{16}$): The definition of the finite field $\mathbb{F}_{16}$ is now given. The field $\mathbb{F}_{16}$ has elements,

$$\{0, 1, \alpha, \alpha^2, \ldots, \alpha^{14}\}.$$

It is easy to see that α is a primitive element in $\mathbb{F}_{16}$ and the polynomial $p(x) = x^4 + x + 1$ is a primitive polynomial. Each element of $\mathbb{F}_{16}$ can be represented as a polynomial in the ring $\mathbb{F}_2[x]/p(x)$. This ring has coefficients in $\mathbb{F}_2$ and any polynomial in the ring cannot be a multiple of $p(x)$. Table 2.1 shows the elements of $\mathbb{F}_{16}$ in different representations. The coefficients of the ring $\mathbb{F}_2[x]/p(x)$ map to an m-dimensional vector space $\mathbb{F}_2^m$ which can also be used to represent $\mathbb{F}_{q^m}$.

2.1.1 Subfields and Conjugacy Classes

2.3 ***Definition (Conjugacy Class).*** *A conjugacy class of an element β of a finite field $\mathbb{F}_{q^m}$ is given as the set of distinct elements,*

$$C(\beta) = \{\beta, \beta^q, \beta^{q^2}, \ldots., \beta^{q^{(e-1)}}\} \quad \beta \in \mathbb{F}_{q^m}$$

where e is the smallest positive integer such that $\beta^{q^e} = \beta$.

Where β is used as a representative of the its conjugacy class. Conjugacy classes partition $\mathbb{F}_{q^m}$ into sets of size r and $r|m$. As an example consider the conjugacy classes of $\mathbb{F}_{16}$ over $\mathbb{F}_2$,

$\mathbb{F}_{2^4}$	$\mathbb{F}_2[x]/(x^{2^4-1})$	$\mathbb{F}_2[x]/p(x)$	$\mathbb{F}_2^4$
0	0	0	[0,0,0,0]
1	1	1	[0,0,0,1]
α	x	x	[0,0,1,0]
α^2	x^2	x^2	[0,1,0,0]
α^3	x^3	x^3	[1,0,0,0]
α^4	x^4	$x+1$	[0,0,1,1]
α^5	x^5	x^2+x	[0,1,1,0]
α^6	x^6	x^3+x^2	[1,1,0,0]
α^7	x^7	x^3+x+1	[1,0,1,1]
α^8	x^8	x^2+1	[0,1,0,1]
α^9	x^9	x^3+x	[1,0,1,0]
α^{10}	x^{10}	x^2+x+1	[0,1,1,1]
α^{11}	x^{11}	x^3+x^2+x	[1,1,1,0]
α^{12}	x^{12}	x^3+x^2+x+1	[1,1,1,1]
α^{13}	x^{13}	x^3+x^2+1	[1,1,0,1]
α^{14}	x^{14}	x^3+1	[1,0,0,1]

Table 2.1: Finite Field $\mathbb{F}_{16}$

$$\{\{1\},$$
$$\{\alpha,\alpha^2,\alpha^4,\alpha^8\},$$
$$\{\alpha^3,\alpha^6,\alpha^{12},\alpha^9\},$$
$$\{\alpha^5,\alpha^{10}\},$$
$$\{\alpha^7,\alpha^{14},\alpha^{13},\alpha^{11}\}\}.$$

Conjugacy classes are also called cyclotomic cosets. A polynomial with all the members of the conjugacy class of the primitive element β of the finite field $\mathbb{F}_{q^m}$ as roots is a primitive polynomial in the subfield $\mathbb{F}_q$ (Lidl and Niederreiter, 1986). Thus to find the primitive polynomial of any subfield $\mathbb{F}_q$ of $\mathbb{F}_{q^m}$ it is sufficient to obtain the conjugacy class of a primitive element of $\mathbb{F}_{q^m}$. Consider the conjugacy classes of $\mathbb{F}_{16}$ over its subfield $\mathbb{F}_4$ with $m=2$,

$$\{\{1\},$$
$$\{\alpha,\alpha^4\},$$
$$\{\alpha^2,\alpha^8\},$$
$$\{\alpha^3,\alpha^{12}\},$$
$$\{\alpha^5\},$$
$$\{\alpha^6,\alpha^9\},$$
$$\{\alpha^7,\alpha^{13}\},$$
$$\{\alpha^{10}\},$$
$$\{\alpha^{11},\alpha^{14}\}\}.$$

The elements of $\mathbb{F}_4 \subset \mathbb{F}_{16}$ are,

$$\{0, 1, \alpha^5, \alpha^{10}\}$$

and α is a primitive element of $\mathbb{F}_{16}$. The polynomial with roots $\{\alpha, \alpha^4\}$ is $x^2 + x + \alpha^5$ and is primitive over $\mathbb{F}_4$.

In Table 2.1 it was shown that an extension field $\mathbb{F}_{2^4}$ can be represented by a 4-dimensional vector $\mathbb{F}_2^4$. The finite field $\mathbb{F}_{16}$ is now defined as a vector space $\mathbb{F}_4^2$ using the primitive polynomial $p(x) = x^2 + x + \alpha^5$ in the same manner. Table 2.2 shows the finite field $\mathbb{F}_{16}$ represented as $\mathbb{F}_4^2$ using the primitive polynomial $p(x) = x^2 + x + \alpha^5$.

$\mathbb{F}_{2^4}$	$\mathbb{F}_4[x]/(x^{4^2-1})$	$\mathbb{F}_4[x]/p(x)$	$\mathbb{F}_4^2$
0	0	0	$[0,0]$
1	1	1	$[0,1]$
α	x	x	$[1,0]$
α^2	x^2	$x+\alpha^5$	$[1,\alpha^5]$
α^3	x^3	$x+\alpha^{10}$	$[1,\alpha^{10}]$
α^4	x^4	$x+1$	$[1,1]$
α^5	x^5	α^5	$[0,\alpha^5]$
α^6	x^6	$\alpha^5 x$	$[\alpha^5,0]$
α^7	x^7	$\alpha^5 x+\alpha^{10}$	$[\alpha^5,\alpha^{10}]$
α^8	x^8	$x+\alpha^{10}$	$[1,\alpha^{10}]$
α^9	x^9	$\alpha^5 x+\alpha^5$	$[\alpha^5,\alpha^5]$
α^{10}	x^{10}	α^{10}	$[0,\alpha^{10}]$
α^{11}	x^{11}	$\alpha^{10} x$	$[\alpha^{10},0]$
α^{12}	x^{12}	$\alpha^{10} x+1$	$[\alpha^{10},1]$
α^{13}	x^{13}	$\alpha^5 x+1$	$[\alpha^5,1]$
α^{14}	x^{14}	$\alpha^{10} x+\alpha^{10}$	$[\alpha^{10},\alpha^{10}]$

Table 2.2: Finite Field $\mathbb{F}_{16}$

2.2 Linear Codes

2.4 ***Definition (Linear Code).*** *A linear code is an n-dimensional vector space $\mathscr{C} \subset \mathbb{F}_q^n$ that can be defined with a basis consisting of k-linearly independent members.*

This vector space is known as a *code space* and consists of $q^k = |\mathscr{C}|$ distinct vectors of length n. A linear code is said to have length n, dimension k and rate $r = \frac{k}{n}$. A matrix $\mathbf{G}$ consisting of k linearly independent members of $\mathscr{C}$ is called the generator matrix of $\mathscr{C}$. An encoding operation can be seen as a map from a message space $\mathbb{F}_q^k$ to the code space $\mathscr{C} \subset \mathbb{F}_q^n$. Consider the map,

$$\gamma : \mathbb{F}_q^m \mapsto \mathscr{C}$$
$$\gamma(\mathbf{m}) = \mathbf{c} \qquad \mathbf{m} \in \mathbb{F}_q^k,\ \mathbf{c} \in \mathscr{C}$$

which represents an encoding operation. Encoding, carried out with the **G** matrix using matrix multiplication is,

$$\mathbf{c} = \mathbf{mG}.$$

2.5 ***Definition (Generator Matrix).*** *A generator matrix of a linear code $\mathscr{C}$ is a $k \times n$ matrix of rank k whose rows are members of $\mathscr{C}$.*

Another important matrix associated with a linear code is the parity check matrix, **H**.

2.6 ***Definition (Dual Code).*** *A parity check matrix* **H** *is an $(n-k) \times n$ matrix of rank $n-k$ which has the property,*

$$\mathbf{cH}^T = \mathbf{0}$$

for every $\mathbf{c} \in \mathscr{C}$ where T is the transpose operator.

The parity check matrix is used to "test" for codewords of $\mathscr{C}$. The parity check matrix of a linear code is simply the null space of its generator matrix and is defined as, $\mathbf{H}^T$ in the equation,

$$\mathbf{GH}^T = \mathbf{0}.$$

Let $[n,k,d]_q$ denote a linear code with length n, dimension k and distance d defined in a field of size q.

2.7 ***Definition.*** *A dual code $\mathscr{C}^\perp$ of a linear code $\mathscr{C}$ is a code which has the parity check matrix H of $\mathscr{C}$ as its generator matrix. Additionally for any two codewords $\mathbf{c} \in \mathscr{C}$ and $\bar{\mathbf{c}} \in \mathscr{C}^\perp$,*

$$\mathbf{c} \cdot \bar{\mathbf{c}} = \mathbf{0}$$

where $\cdot$ denotes the component-wise multiplication of vectors or dot product.

The dual code of the code $\mathscr{C}$ with parameters $[n,k,d]_q$ has length n, dimension $k^\perp = n-k$ and minimum distance denoted by $d^\perp$.

2.2.1 Properties of Linear Codes

2.8 ***Definition (Codeword Weight).*** *The number of nonzero elements of a codeword* $\mathbf{c} \in \mathscr{C}$ *is called its weight. Formally,*

$$weight(\mathbf{c}) = |\{i : c_i \neq 0,\ i = [0 \ldots n-1]\}| \qquad \mathbf{c} = (c_0, \ldots, c_{n-1})$$

2.9 ***Definition (Hamming Distance).*** *The Hamming distance between any two codewords* $\mathbf{x}, \mathbf{y} \in \mathscr{C}$ *is given as the weight of their dot product. Formally,*

$$d(\mathbf{x}, \mathbf{y}) = weight(\mathbf{x} \cdot \mathbf{y})$$

2.10 ***Definition (Minimum Weight).*** *The minimum distance or weight of a linear code* $\mathscr{C}$ *is smallest distance between any two distinct codewords in* $\mathscr{C}$.

$$d(\mathscr{C}) = min(\{d(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathscr{C}\})$$

The minimum distance of a linear code is a useful indicator of the quality of that code. It is desirable to have linear codes with as large a minimum distance as possible for a given length, dimension and field size.

2.11 ***Definition (Weight Distribution).*** *Suppose* A_i *is defined as the number of codewords in a code* $\mathscr{C}$ *of weight equal to* i

$$A_i = |\{weight(\mathbf{c}) = i \,|\, \mathbf{c} \in \mathscr{C}\}|$$

then the sequence

$$[A_0, A_1, \ldots, A_n]$$

is called the weight distribution of the code $\mathscr{C}$.

If $M = q^k$ is the number of codewords in $\mathscr{C}$ then

$$M = \sum_{i=0}^{n} A_i.$$

The weight distribution of a code is useful in predicting its performance under maximum likelihood decoding. Also if the weight distribution of a code is known the MacWilliams identity (MacWilliams and Sloane, 1983) can be used to obtain the weight distribution of its dual code.

2.3 Generic Code Constructions

This section is concerned with the construction of new codes from existing ones. Henceforth a linear code is assumed to have parameters $[n,k,d]_q$.

2.3.1 Modifying The Length

2.3.1.1 Padding

A linear code can be lengthened if every codeword is padded with a zero symbol. The result is an $[n+1,k,d]_q$ code.

2.3.1.2 Overall Parity Check

A linear code can be extended to an $[n+1,k,d+1]_q$ code if $q=2$ and d is odd. Every codeword $\mathbf{c} \in \mathscr{C}$ is extended as such,

$$(c_0,\ldots,c_{n-1},\sum_{i=0}^{n-1} c_i)$$

with $c_n = \sum_{i=0}^{n-1}$. For cases where $q \neq 2$ adding an overall parity check may or may not increase the distance to $d+1$. For special cases however it has been shown (Simonis, 2000) that it is possible to increase the distance to $d+1$.

2.3.1.3 Puncturing

A linear code can be punctured to an $[n-l,k,\geq d-l]_q$ provided that $|l| < d$. Puncturing involves removing l columns of the generator matrix of a code.

2.3.1.4 Construction X

2.12 ***Definition (Subcode).*** *A subcode $\mathscr{C}_2$ of a linear code $\mathscr{C}_1$ is a code that has all its codewords in $\mathscr{C}_1$.*

$$\mathscr{C}_2 \subset \mathscr{C}_1$$

The subcode $\mathscr{C}_2$ has parameters $[n,<k,\geq d]_q$.

The difference between the dimension of a code and the dimension of its subcode is called co-dimension.

2.1 Theorem (Construction X (Sloane et al., 1972)). *If a linear code $\mathscr{C}_1$ with parameters $[n, k_1, d_1]$ has a subcode $\mathscr{C}_2$ with parameters $[n, k_2, d_2]$, then $\mathscr{C}_1$ is extendable to a code with parameters $[n+\acute{n}, k_1, \min\{d_1+\delta, d_2\}]$ using some auxiliary code $[\acute{n}, k_1-k_2, \delta]$.*

Consider a linear code $\mathscr{C}_1$ with parameters $[n, k_1, d_1]$ with generator matrix $\mathbf{G}_1$ which can be represented as,

$$\mathbf{G}_1 = \begin{bmatrix} \mathbf{G}_2 \\ \mathbf{G} \end{bmatrix}$$

where $\mathbf{G}_2$ is the generator matrix of a subcode $\mathscr{C}_2$ with parameters $[n, k_2, d_2]$. Suppose the auxiliary code $\mathscr{C}_3$ with parameters $[\acute{n}, k_1-k_2, \delta]$ has generator matrix $\mathbf{G}_3$, then the generator matrix (Grassl, 2006) of a code obtained from construction X using these three codes has generator matrix,

$$\acute{\mathbf{G}} = \begin{bmatrix} \mathbf{G}_2 & \mathbf{0} \\ \mathbf{G} & \mathbf{G}_3 \end{bmatrix}.$$

In some cases the subcode $\mathscr{C}_2$ has a length shorter than the length of the supercode $\mathscr{C}_1$. This is the case when $\mathscr{C}_2$ is obtained from $\mathscr{C}_1$ by shortening. In which case the code $\mathscr{C}_2$ is padded with zeros in the shortened coordinates.

2.1 Corollary. *If a linear code $\mathscr{C}_1$ with parameters $[n_1, k_1, d_1]$ has a subcode $\mathscr{C}_2$ with parameters $[n_2, k_2, d_2]$ with $n_2 \leq n_1$, then $\mathscr{C}_1$ is extendable to a code with parameters $[n_1+\acute{n}, k_1, \min\{d_1+\delta, d_2\}]$ using some auxiliary code $[\acute{n}, k_1-k_2, \delta]$.*

Other extensions methods worthy of mention are constructions X3, X3a and X3u which use three nested codes,construction X4 which uses two pairs of nested codes and constructions X6, X6a and X6u which use 6 nested codes. (See Brouwer, 1998; MacWilliams and Sloane, 1983) for details of these constructions.

2.3.2 Modifying The Dimension

2.3.2.1 Subcodes

Given a code $\mathscr{C}$ with parameters $[n, k, d]$ it is possible to form an l co-dimensional subcode by deleting l rows of the generator matrix of the code $\mathscr{C}$. The resulting code has parameters $[n, k-l, \geq d]$.

2.3.2.2 Shortening

A code can be shortened by deleting l information coordinates with $l < k$. The shortened code has parameters $[n-l, k-l, \geq d]_q$. These l deleted coordinates need to be a subset of an information set.

2.13 ***Definition (Information Sets).*** *If* $\mathbf{G}$ *is the generator matrix of a linear code, then an information set is a set of coordinates of any* k *linearly independent columns of* $\mathbf{G}$.

Shortening can be accomplished by deleting l independent columns of the generator matrix $\mathbf{G}$ as well as l rows. Deleting l independent columns of the parity check matrix $\mathbf{H}$ also has the same effect. In order to state Theorem 2.2, the definition of a *support* is first given.

2.14 ***Definition (Support of a Codeword).*** *Let* $\mathbf{c} \in \mathscr{C}$ *a codeword of* $\mathscr{C}$ *then the support of* $\mathbf{c} = (c_0, \ldots, c_{n-1})$ *is defined as ,*

$$supp(\boldsymbol{c}) = \{i : i \in \{0, \ldots, n-1\} \mid c_i \neq 0\}.$$

2.2 ***Theorem (Construction Y1 , from (MacWilliams and Sloane, 1983)).*** *If the dual of the code* $\mathscr{C}$ *with parameters* $[n,k,d]$ *has a codeword* $\acute{\mathbf{c}}$ *of minimum weight* $\acute{d}$*, then deleting the columns of the parity check matrix of* $\mathscr{C}$ *corresponding to the support of* $\acute{\mathbf{c}}$ *produces a shortened code with parameters* $[n-\acute{d}, k-\acute{d}+1, d]$.

2.3.3 Subfield Constructions

Given a linear code $\mathscr{C}$ defined in some extension field $\mathbb{F}_{q^m}$ it is possible to obtain codes from $\mathscr{C}$ having elements in a subfield $\mathbb{F}_q$ of $\mathbb{F}_{q^m}$. There are three basic ways to do this; by constructing a subfield image code , by constructing a subfield subcode or by constructing a trace code.

2.3.3.1 Subfield Image Construction

In the beginning of this chapter a method of representing finite fields was shown. Another way of representing finite fields is by using matrices. Let $p(x)$ be a primitive polynomial of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. The companion matrix of a polynomial $f(x) =$

$a_0 + a_1 x + \cdots + a_{m-1}x^{m-1} + x^m$ is defined as an $m \times m$ matrix given as,

$$C = \begin{bmatrix} 0 & 0 & \cdots & 0 & -a_0 \\ 1 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & \cdots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & 1 & -a_{m-1} \end{bmatrix}$$

and satisfies $f(C) = \mathbf{0}$ where $\mathbf{0}$ is an $m \times m$ matrix with all zero entries (Lidl and Niederreiter, 1986; MacWilliams and Sloane, 1983). Let C be the companion matrix of the primitive polynomial $p(x)$, then there is a one to one mapping between the elements of the finite field $\mathbb{F}_{q^m}$ and the set

$$\{\mathbf{0}\} \cup \{C^i : i \in [1 \ldots (q^m - 1)]\}.$$

The map σ_m is given by,

$$\begin{aligned} &\sigma_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^{m \times m} \\ &\sigma_m(\alpha^j) = C^j, \qquad \alpha^j \in \mathbb{F}_{q^m} \setminus \{0\} \\ &\sigma_m(0) = \mathbf{0} \end{aligned}$$

where α is the primitive element of $\mathbb{F}_{q^m}$. This map is denoted as σ_m,

$$\sigma_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^{m \times m}.$$

In summary each symbol in $\mathbb{F}_{q^m}$ can be represented by a unique $m \times m$ matrix.

Example 2.2: Consider the finite field $\mathbb{F}_8$ defined with the primitive polynomial $p(x) = x^3 + x + 1$. The companion matrix of $p(x)$ is,

$$C = \begin{bmatrix} 001 \\ 101 \\ 010 \end{bmatrix}.$$

It is then straightforward to map,

$$0 \mapsto \begin{bmatrix} 000 \\ 000 \\ 000 \end{bmatrix} \qquad \text{and} \qquad \alpha^i \mapsto C^i \ , \ i = [1, \ldots, 7]$$

where α is the primitive element of $\mathbb{F}_8$.

It is now possible to give a construction of subfield image codes. Let $\mathbf{G}$ be the generator matrix of a linear code $\mathscr{C}$ with parameters $[n, k, d]_{q^m}$ defined in the finite field

$\mathbb{F}_{q^m}$. The matrix $\mathbf{G}$ can be defined as,

$$\mathbf{G} = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}.$$

A generator matrix of the subfield image code using $\mathbf{G}$ and the map σ_m can be formed. The subfield image code $\mathscr{D}$ has parameters $[nm, mk, \geq d]_q$ and generator matrix $\acute{\mathbf{G}}$,

$$\acute{\mathbf{G}} = \begin{bmatrix} \sigma_m(g_{0,0}) & \sigma_m(g_{0,1}) & \cdots & \sigma_m(g_{0,n-1}) \\ \sigma_m(g_{1,0}) & \sigma_m(g_{1,1}) & \cdots & \sigma_m(g_{1,n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_m(g_{k-1,0}) & \sigma_m(g_{k-1,1}) & \cdots & \sigma_m(g_{k-1,n-1}) \end{bmatrix}.$$

2.3.3.2 Subfield Subcode Construction

2.15 Definition (Subfield Subcode). *A subfield subcode $\mathscr{C}|_{\mathbb{F}_q}$ of a code $\mathscr{C}$ defined in $\mathbb{F}_{q^m}$ consists of all those codewords in $\mathscr{C}$ that have all their elements in the subfield $\mathbb{F}_q$.*

It is possible to construct the parity check matrix of a subfield subcode from the parity check matrix of the code $\mathscr{C}$ defined in $\mathbb{F}_{q^m}$. First the map π_m is defined. In Section 2.1 it was shown that there is a one to one mapping between the elements of $\mathbb{F}_{q^m}$ and the quotient ring $\mathbb{F}_q[x]/p(x)$. Consequently there is also a one to one mapping between elements of $\mathbb{F}_{q^m}$ and the vector space $\mathbb{F}_q^m$ formed from the coefficients of the elements in $\mathbb{F}_q[x]/p(x)$. The map π_m is defined as,

$$\begin{aligned} &\pi_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^m \\ &\quad \pi_m(\beta) = (a_0, a_1, \ldots, a_{m-1})^{\mathrm{T}}, \quad \beta \in \mathbb{F}_{q^m},\ a_i \in \mathbb{F}_q, \end{aligned}$$

where T is the transpose operator, which maps elements of $\mathbb{F}_{q^m}$ to $\mathbb{F}_q^m$. Suppose $\mathbf{H}$ is the parity check matrix of the code $\mathscr{C}$ in $\mathbb{F}_{q^m}$,

$$H = \begin{bmatrix} h_{0,0} & h_{0,1} & \cdots & h_{0,n-1} \\ h_{1,0} & h_{1,1} & \cdots & h_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{r-1,0} & h_{r-1,1} & \cdots & h_{r-1,n-1} \end{bmatrix}$$

with redundancy $r = n-k$, then the parity check matrix of the subfield subcode $\mathscr{C}|_{\mathbb{F}_q}$ is given (MacWilliams and Sloane, 1983), by,

$$\tilde{\mathbf{H}} = \begin{bmatrix} \pi_m\left(h_{0,0}\right) & \pi_m\left(h_{0,1}\right) & \cdots & \pi_m\left(h_{0,n-1}\right) \\ \pi_m\left(h_{1,0}\right) & \pi_m\left(h_{1,1}\right) & \cdots & \pi_m\left(h_{1,n-1}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \pi_m\left(h_{r-1,0}\right) & \pi_m\left(h_{r-1,1}\right) & \cdots & \pi_m\left(h_{r-1,n-1}\right) \end{bmatrix}.$$

A subfield subcode has parameters $[n, \geq n-mr, \geq d]_q$.

2.3.3.3 Trace Construction

2.16 *Definition (Trace of an element).* *The trace of an element $\beta \in \mathbb{F}_{q^m}$ is defined as,*

$$T_m(\beta) = \sum_{i=0}^{m-1} \beta^{q^i}.$$

Suppose $\mathscr{C}$ is a linear $[n,k,d]_{q^m}$ code defined in the finite field $\mathbb{F}_{q^m}$ with codewords $\mathbf{c} = (c_0, \ldots, c_{n-1})$ its corresponding trace code $T_m(\mathscr{C})$ consists of all codewords of the form,

$$(T_m(c_0), T_m(c_1), \ldots, T_m(c_{n-1})) \qquad \mathbf{c} \in \mathscr{C}.$$

The trace code $T_m(\mathscr{C})$ has parameters $[n, \geq k, \leq d]_q$ code. An interesting relationship between subfield subcodes and trace codes was given by Delsarte (1975).

2.3 *Theorem (Delsarte).* *The dual of a subfield subcode is the trace of the dual of the original code $\mathscr{C}$ defined in $\mathbb{F}_{q^m}$,*

$$\left(\mathscr{C}|_{\mathbb{F}_q}\right)^{\perp} = T_m((\mathscr{C})^{\perp})$$

2.3.4 Code Concatenation

A linear code $\mathscr{C}_1$ with parameters $[n_1,k,d_1]_{q^m}$ can be concatenated with a linear code $\mathscr{C}_2$ with parameters $[n_2, \frac{m}{r}, d_2]_{q^r}$ code provided that r divides m. $\mathscr{C}_1$ is called

the inner code while $\mathscr{C}_2$ is called the outer code. Let us define the map,

$$\psi_p : \mathbb{F}_{q^r}^{p\times p} \mapsto \mathscr{C}_2^{p\times n2}$$
$$\psi_p(\mathbf{M}) = \mathbf{M}\mathbf{G}_2 = \mathbf{N} \qquad \mathbf{M} \in \mathbb{F}_{q^r}^{p\times p}$$

which describes multiplication of a $p \times p$ matrix $\mathbf{M}$ with the generator matrix $\mathbf{G}_2$ of $\mathscr{C}_2$. The result of this multiplication is a $p \times n_2$ matrix $\mathbf{N}$ with each row a codeword of $\mathscr{C}_2$. Let $p = \frac{r}{m}$. A concatenated code has generator matrix,

$$\acute{\mathbf{G}} = \begin{bmatrix} \psi_p\left(\sigma_m\left(g_{0,0}\right)\right) & \psi_p\left(\sigma_m\left(g_{0,1}\right)\right) & \cdots & \psi_p\left(\sigma_m\left(g_{0,n-1}\right)\right) \\ \psi_p\left(\sigma_m\left(g_{1,0}\right)\right) & \psi_p\left(\sigma_m\left(g_{1,1}\right)\right) & \cdots & \psi_p\left(\sigma_m\left(g_{1,n-1}\right)\right) \\ \vdots & \vdots & \ddots & \vdots \\ \psi_p\left(\sigma_m\left(g_{k-1,0}\right)\right) & \psi_p\left(\sigma_m\left(g_{k-1,1}\right)\right) & \cdots & \psi_p\left(\sigma_m\left(g_{k-1,n-1}\right)\right) \end{bmatrix}$$

when $\mathbf{G}_1$ is the generator matrix of the code $\mathscr{C}_1$ defined as,

$$\mathbf{G}_1 = \begin{bmatrix} g_{0,0} & g_{0,1} & \cdots & g_{0,n-1} \\ g_{1,0} & g_{1,1} & \cdots & g_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}.$$

The map σ_m is as previously defined. A concatenated code has parameters $[n_1 n_2, kp, \geq d_1 d_2]_{q^r}$.

2.4 Channel Models

In order for channels to be analysed, models are built that encapsulate the behaviour of the channel. An important channel model is the discrete memoryless channel (DMC) in which transmitted symbols are corrupted independently of each other. Given input alphabet $A = \{a_0, a_1, \ldots, a_{q-1}\}$ and output alphabet $B = \{b_0, b_1, \ldots, b_{r-1}\}$ for the DMC a set of qr conditional probabilities arise,

$$P(B = b_i | A = a_j) \equiv P(y_i | x_j) \quad i = 0, \ldots, q-1 \quad j = 0, \ldots, r-1$$

Any sequence of n symbols from the input alphabet A denoted as $u_0, \ldots, u_{n-1}$ and a corresponding sequence of output symbols $v_0, \ldots, v_{n-1}$ form B, the DMC has joint conditional probability ,

$$P(B = v_1, \ldots, B = v_{n-1} | A = u_1, \ldots, A = u_{n-1}) = \prod_{k=1}^{n} P(B = v_k | A = u_k)$$

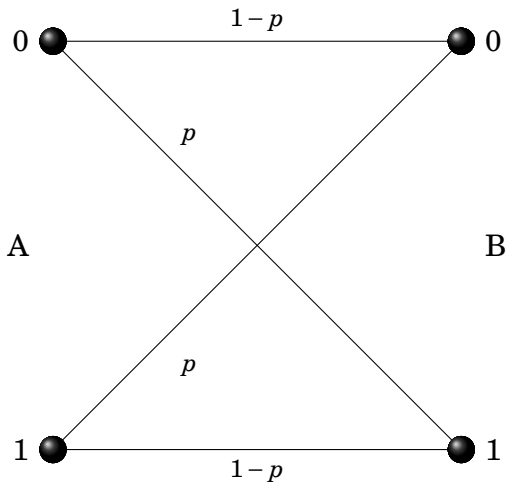

Fig. 2.1: Pictorial representation of the BSC

with the right hand side of the equation showing the memoryless nature of the channel. A particular type of DMC is the binary symmetric channel (BSC). For the BSC, $q = r = 2$ and $A = B = \{0, 1\}$. Figure 2.1 shows a pictorial representation of the BSC with channel probability p. The AWGN channel is a discrete-time memoryless channel with discrete input alphabet $A = \{a_0, \ldots, a_{q-1}\}$ and real output alphabet $B = (-\infty, \infty)$ where

$$B = A + X$$
$$b_i = a_i + x_i$$

where X is Gaussian random variable with variance σ^2 and a zero mean (Proakis, 2008). Another important channel model is the erasure channel which is also memoryless. The binary erasure channel (BEC) is a specific type of erasure channel. A BEC is a binary input channel with the possibility of an erasure at the output. The BEC is shown pictorially in Figure 2.2 with a channel probability p where ? denotes an erasure. The nonbinary erasure channel is similar to the BEC except the input and output symbols can take nonbinary values. The AWGN is an accurate communication link for satellite and communication channels where the noise contribution is due to thermal or intergalactic noise while the erasure channel is used to model packet based networks.

2.5 Computing Minimum Distances

Determining the minimum distance of a code is a difficult problem and has been shown to be NP-complete (Vardy, 1997) for linear codes in arbitrary sized finite fields. For moderate length codes one may use a brute force approach to find the minimum distance. Depending on the rate of a code it is possible to either use the parity check matrix $\mathbf{H}$ or the generator matrix $\mathbf{G}$ to compute the minimum distance

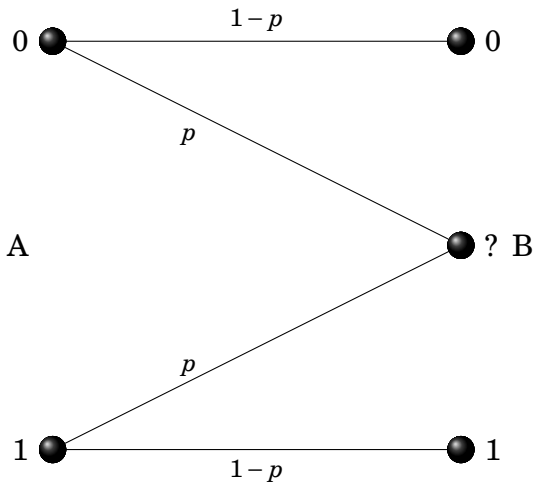

Fig. 2.2: Pictorial representation of the BEC

using an exhaustive search. Using the generator matrix, an exhaustive search involves encoding all q^k codewords and selecting codeword/codewords with the least minimum weight. For codes with relatively small dimensions (low rate) an exhaustive search is feasible[1] using this method. For codes with a large dimension (high rate) the number of computations increases for this method. To describe exhaustive search using the parity check matrix of a code Theorem 2.4 is stated.

2.4 *Theorem (From (MacWilliams and Sloane, 1983)).* *A code with minimum distance d has every combination of $d-1$ or less columns of its parity check matrix linearly independent.*

The task of finding the minimum weight of a code then becomes checking all $\sum_{i=2}^{d-1}\binom{n}{i}$ columns of the parity check matrix $\mathbf{H}$ for linear independence. Once a set of columns d linearly dependent columns are found the search is complete. However even for lengths up to $n \leq 256$ using these brute force approaches can be impractical (especially if the rate $\frac{k}{n} \approx 0.5$) using a common personal computer. Probabilistic methods can be used compute the minimum distance of a linear codes to a certain probability of accuracy. Probabilistic methods typically truncate the minimum distance search after a chosen number of computations and hence cannot guarantee that their output is indeed the true minimum distance of the code. These methods are far less time consuming and are especially useful in verifying the minimum distance of constructed codes for which a proven lower bound is available. They can also be used to obtain low weight codewords of the code, an attribute that can be exploited to obtain a partial weight distribution of the code or to construct low density parity check matrices. A probabilistic algorithm for computing the minimum weight of a linear code was presented in (Canteaut and Chabaud, 1998). This

[1] Feasibility as used here means that an exhaustive search is possible in a reasonable amount of time using a single computer.

method is a modification of a probabilistic method by Stern (Stern, 1989). In this section details of a probabilistic algorithm of computing minimum distance of linear codes presented by Tomlinson et al. (2007) are given.
It can be deduced from Theorem 2.4 that the parity check matrix of a linear code with minimum weight d has every possible $s \times s$ submatrix nonsingular provided $s < d$. It is well known in linear algebra that the nonsingularity of a matrix formed from the coefficients of a set of linear homogeneous equations guarantees that the equations can be solved for unknowns. This observation forms the basis of the probabilistic method of computing minimum distance using erasures.

2.5.1 Probabilistic Method of Finding Minimum Distance Using Erasures

If the minimum number of erasures that a code cannot correct is s then its minimum distance is $s+1$ and the erasure pattern that meets this criterion can be solved to find the minimum weight codeword. Any set of erasures on transmitted codewords can be corrected as long as the number erasures does not exceed the erasure correcting capacity of the code and the coordinates of the erasures in the erasure pattern correspond to the linearly independent columns of the parity check matrix. The latter criterion is needed to ensure that the erasures can be solved using a set of simultaneous equations and that there are no dependencies between erasure components, leading to more unknowns than there are equations. This means that coordinates of correctable erasure patterns are synonymous with the linearly independent columns of the parity check matrix while coordinates of uncorrectable erasure patterns are synonymous with the linearly dependent columns of a the parity matrix. Provided that the criterion for the erasure patterns corresponding to the positions of the linearly independent columns parity matrix is met, the simultaneous equations formed by multiplication with the codeword and the parity matrix can be expressed in reduced Gaussian form . For z pattern of erasures occurring in the first z columns of the $\mathbf{H}$ matrix assumed here to be linearly independent (although in practise can be in any order), then in reduced Gaussian form

$$\begin{aligned}
e_0 &= h_{0,0}x_0 + h_{0,1}x_1 + \cdots\cdots + h_{0,n-z-1}x_{n-z-1} \\
e_1 &= h_{1,0}x_0 + h_{0,1}x_1 + \cdots\cdots + h_{0,n-z-1}x_{n-z-1} \\
&\vdots \\
&\vdots \\
e_{z-1} &= h_{1,0}x_0 + h_{0,1}x_1 + \cdots\cdots + h_{0,n-z-1}x_{n-z-1}
\end{aligned} \tag{2.1}$$

The rest of the $n-k-z$ equations in which no erasures are present are

$$h_{z,0}x_0 + h_{z,1}x_1 + \cdots\cdots + h_{z,n-z-1}x_{n-z-1} = 0$$
$$h_{z+1,0}x_0 + h_{z+1,1}x_1 + \cdots\cdots + h_{z+1,n-z-1}x_{n-z-1} = 0$$
$$\vdots$$
$$\vdots$$
$$h_{n-k-1,0}x_0 + h_{n-k-1,1}x_1 + \cdots\cdots + h_{n-k-1,n-z-1}x_{n-z-1} = 0$$

Code	Field size, q	Length, n	Dimension, k	Minimum distance d_{min}	Time, seconds	Trials
Hermitian	16	60	45	10	658	5926319
Hermitian	16	50	40	5	0	174
Hermitian	4	8	2	6	0	1
Klein	8	21	15	4	0	286

Table 2.3: Timings for different codes

From (2.1) it is possible to solve for each erasure. Now consider the case of the lowest weight codeword with a weight equal to the minimum distance of the code with erasures occurring in positions of the codeword where elements are non-zero, this represents w erasures. It is well known that from the multiplication of codewords with the transpose parity check matrix $c\mathbf{H}^T = 0 \ \forall \ c \in C$ every multiplication $c\mathbf{H}^T$ is a linear combination of the columns of $\mathbf{H}$ and multiplication with the least weight codeword implies that there is a set of linearly dependent columns in $\mathbf{H}$ such that the set size is minimal [2] and is equal to the minimum distance. Therefore the coordinates of the erasures of this lowest weight codeword correspond to the linearly dependent columns of the $\mathbf{H}$ matrix and this erasure pattern cannot be solved. However for the same codeword there is an erasure pattern of $w-1$ erasures which now correspond to the linearly independent coordinates of $\mathbf{H}$ and can be solved. If these $w-1$ erasures cannot be solved then it means that the coordinates correspond to linearly dependent columns of $\mathbf{H}$ and a codeword of weight $w-1$ exists, which is a contradiction. The steps to finding the minimum distance via the method proposed by (Tomlinson et al., 2007) are

[2]The minimum distance of a code is equal to the least number of linear dependent columns of its parity check matrix.

Algorithm 2.1 Erasure method

Require: H

1: **for** $i = 1 : s$ **do**
2: Choose $n-k$ random columns of the **H** matrix
3: Find the rank r of an $(n-k) \times (n-k)$ submatrix formed from these columns and the rows of **H**
4: Store $(r+1)_i$
5: **end for**
6: Set $d_{min} = min\{(r+1)_i \forall i\}$.

Solving for the erasures is simple over $GF(2)$ since the weight is already known to be $d_{min} = r_{min} + 1$, the $(r_{min}+1)$th erasure can be assumed to be 1 and the rest of the erasures can be solved by back substitution. For the non-binary case over $GF(2^m)$ with $m > 1$, the $(r_{min}+1)$th erasure can be assumed to take each value of the finite field and for each element a codeword will be formed by back substitution of the equations to form a set of $q-1$ codewords. The minimum codeword will be the codeword with the minimum weight from this set. The method is well suited for non-binary codes because it does not test linear dependence of the columns of the parity check matrix but rather checks for the solvability of a pattern of erasures. In fact apart from the increased complexity of non-binary symbol additions and multiplications, the method is quite similar to the case where it is used for binary codes of the same length/redundancy. Another important factor that determines the speed of the algorithm is the behaviour of the random generator used to select the the columns of the **H** matrix which causes the speed of the algorithm to vary with each search. Table 2.3 gives timings for different codes using this probabilistic method . The search was carried out on a computer with a 1.86GHz central processing unit (CPU) processor and 1.987GB of random access memory (RAM) memory.

2.6 Summary

A brief description of finite fields is given. Notions related to finite fields (subfields and conjugacy classes) are also introduced. Berlekamp (1974) in his survey of key papers in the 1974 attributed the first use of finite fields to E. Prange in an unpublished work. Subsequent work by Zierler (1960) and Mattson and Solomon (1961) showed the effectiveness of the theory of finite fields when applied to error correcting codes. The invention of linear codes is attributed to Hamming (1950) and since his discovery researchers in coding theory strove to cosntruct codes with good error correcting capabilities. Aside from creating entirely new classes of good linear codes, researchers have also focused on methods of producing good codes from existing ones. Some of these methods arose from the search for codes with the minimum distance (construction X (Sloane et al., 1972)) while others like concatenation (For-

ney and Costello, 2007) were a result from the quest to meet Shannon's asymptotic bound. From simple modifications like shortening and puncturing to more advanced constructions these methods are used today to construct better codes and improve communication performance.

Part II

Algebraic Codes For Error Correction

3. RS, BCH AND GOPPA CODES

3.1 Introduction

Binary BCH were discovered by Bose and Chaudhuri (1960) and independently by Hocquenhem (1959). RS codes were later discovered by Reed and Solomon (1960). A year later Gorenstein and Zierler (1961) presented BCH codes with nonbinary symbols. Both BCH and RS codes are ideal-based codes for which every codeword polynomial must have among its roots a certain set of distinct defining elements of a finite field. In general RS codes are considered as a subclass of BCH codes. This view is justified as the defining roots of a BCH code always contain as a subset the defining roots of an RS code. However BCH codes can also be seen as a subclass of RS codes if one considers the fact that a BCH code consists of codewords of an RS code with symbols restricted to a subfield. For a more streamlined categorisation of these codes in relation to other algebraic codes it is better to take the latter view. In which case it is possible to say that BCH codes are subfield subcodes of RS codes.with symbols in the finite field $\mathbb{F}_q$ denoted by $\mathbb{F}_q[x]$.

3.1 *Definition (Ideal, (Cox et al., 2007)).* *A subset $\mathscr{I}$ of $\mathbb{F}_q[x]$, $\mathscr{I} \subset \mathbb{F}_q[x]$ is an ideal if the following conditions are satisfied,*

1. $0 \in \mathscr{I}$
2. $f(x),\ g(x) \in \mathscr{I}$, *then* $f(x)+g(x) \in \mathscr{I}$
3. $f(x) \in \mathscr{I}$ *and* $h(x) \in \mathbb{F}_q[x]$, *then* $h(x)f(x) \in \mathscr{I}$

An ideal can be completely defined by a basis of any l independent generator polynomials

$$< g_1(x), \ldots, g_l(x) > \qquad g_i(x) \in \mathscr{I}$$

Any polynomial in $f(x) \in \mathscr{I}$ can be expressed as the sum

$$f(x) = \sum_{i=1}^{l} h_i(x) g_i(x) \qquad h_i(x) \in \mathbb{F}_q[x]$$

A principal ideal in $\mathbb{F}_q[x]$ is one defined with a basis with a single generator polynomial $g(x)$. In which case any polynomial $f(x) \in \mathbb{F}_q[x]$ is,

$$f(x) = h(x)g(x) \qquad h(x) \in \mathbb{F}_q[x]. \tag{3.1}$$

RS and BCH codes are principal ideals in the univariate ring $\mathbb{F}_q[x]$. Both RS and BCH codes are cyclic codes in that a codeword polynomial results in another codeword polynomial under multiplication by x^t for some t. This property can be readily deduced from the definition of an ideal. The choice of the single generator polynomial $g(x)$ for these codes is restricted by the BCH bound.

3.1 *Theorem (BCH bound,(MacWilliams and Sloane, 1983)).* *Any polynomial $c(x) \in \mathbb{F}_q[x]$ with δ consecutive roots of the finite field $\mathbb{F}_q$ such that,*

$$c(\alpha^b) = c(\alpha^{b+1}) = \cdots = c(\alpha^{b+\delta-1}) = 0$$

for some $\alpha \in \mathbb{F}_q$, has at least $\delta + 1$ nonzero coefficients.

The BCH bound gives the minimum weight of any codeword $c(x)$ of the BCH/RS code. The generator polynomial $g(x)$ for the BCH/RS code must have as a subset of its defining roots a set of consecutive roots $\{\alpha^b, \ldots, \alpha^{b+\delta-1}\}$ so that every codeword has weight at least $\delta + 1$. RS codes have excellent distance properties and achieve the Singleton bound[1]. An RS code with length n and dimension k has minimum distance $d = n - k + 1$. Codes that meet this bound are called maximum distance separable (MDS) codes. RS codes are by no means unique in this sense. A large class of MDS obtainable from RS codes are called generalised Reed Solomon (GRS) codes.

3.2 *Definition (GRS).* *A GRS code is a code that has codewords of the form,*

$$\mathbf{v} \cdot \mathbf{c} = (v_0 c_0, v_1 c_1, \ldots, v_{n-1} c_{n-1})$$

where $\mathbf{c} = (c_0, \ldots, c_{n-1})$ is a codeword of an RS code with parameters $[n, k, d]_q$ defined in the finite field $\mathbb{F}_q$ while the vector template $\mathbf{v} = (v_0, \ldots, v_{n-1})$ has no zero element. GRS codes have parameters $[n, k, d]_q$.

The dual code of a GRS code is also a GRS code albeit defined with a different template vector (MacWilliams and Sloane, 1983).

[1] Singleton bound is $d \leq n - k + 1$

3.3 *Definition (Alternant Codes).* *An alternant code is a subfield subcode of a GRS code. If the GRS code has parameters* $[n, k, n-k+1]_{q^m}$*, alternant code has parameters* $[n, \geq n - m(n-k), \geq n-k+1]_q$.

BCH codes are a subclass of alternant codes for which the template vector $\mathbf{v} = (1, \ldots, 1)$. Another important subclass of alternant codes are Goppa codes. Goppa (1970) introduced this class of codes 10 years after the discovery of BCH codes. Goppa codes were very competitive in terms of good properties and as a class include far more codes than BCH codes. In fact MacWilliams and Sloane (1983) show that narrow-sense primitive BCH codes (a subclass of BCH codes) are also a subclass of Goppa codes.

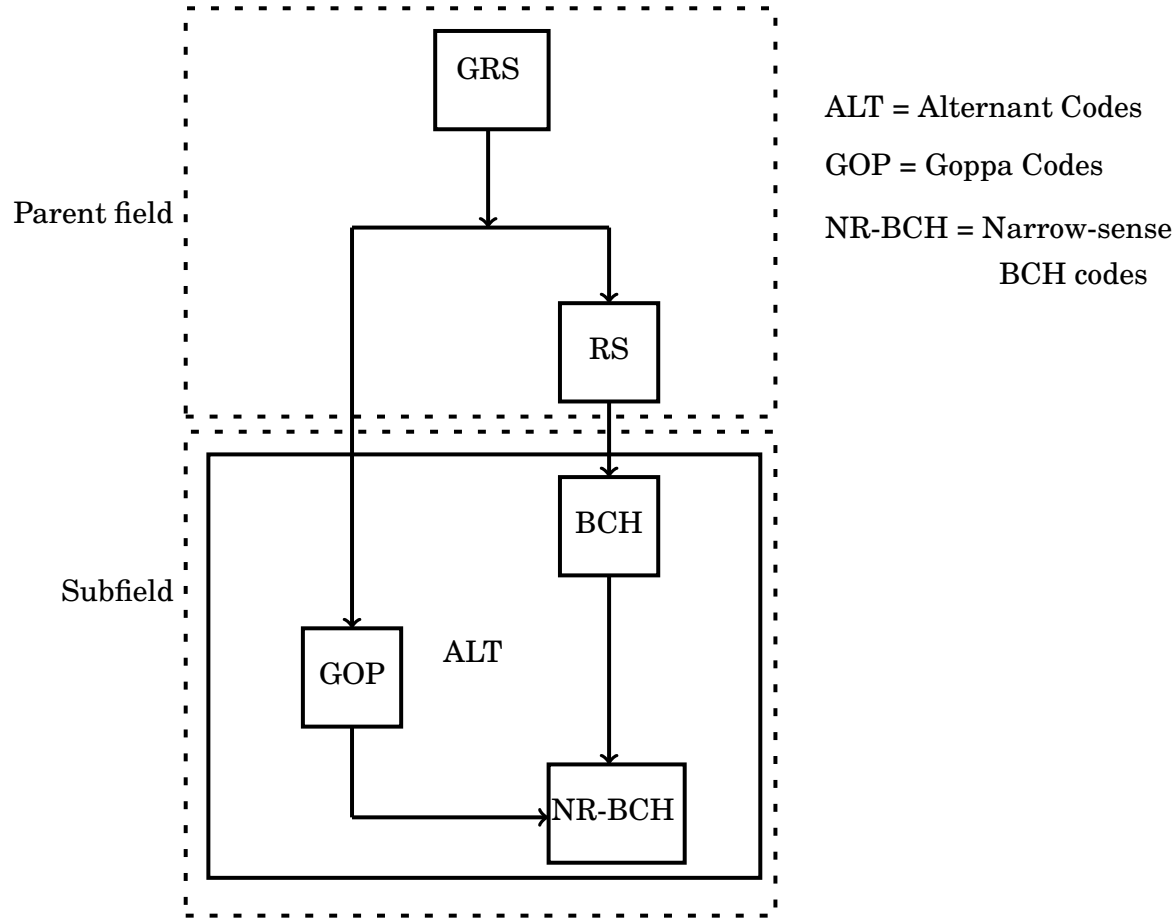

Fig. 3.1: Relationship between algebraic codes

3.2 Reed Solomon Codes

A Reed Solomon code is principal ideal $\mathscr{I}$ in the ring $\mathbb{F}_{q^m}[x]$ with each polynomial $c(x) \in \mathscr{I}$ having distinct cyclically consecutive roots,

$$A = \{\alpha^b, \ldots, \alpha^{b+\delta-1}\} \qquad \alpha \in \mathbb{F}_{q^m}$$

The minimum distance of an RS code is determined by the BCH bound and is $d = \delta + 1$. By definition,

$$c(\alpha^b) = \cdots = c(\alpha^{b+\delta-1}) = 0$$

Suppose the set B is defined as the set $B = \mathbb{F}_{q^m}^* \setminus A$ which contains all elements of the finite field $\mathbb{F}_{q^m}$ except the roots in A and the zero element, it is possible to represent the evaluation of the $c(x)$ for which x takes all the values of the finite field $\mathbb{F}_{q^m}^*$ in the table below. Let $k = |B| = |\mathbb{F}_{q^m}^*| - |A|$.

$c(\alpha^{b-k})$	$c(\alpha^{b-k+1})$	$\cdots$	$c(\alpha^{b-1})$	$c(\alpha^b)$	$\cdots$	$c(\alpha^{b+\delta-1})$
m_0	m_1	$\cdots$	m_{k-1}	0	$\cdots$	0

If $c(x) = c_0 + c_1 x + \cdots + c_{n-1}x^{n-1}$, these evaluations can be represented as a single matrix multiplication,

$$\begin{bmatrix} \alpha^{(b-k)(n-1)} & \alpha^{(b-k)(n-2)} & \cdots & \alpha^{(b-k)} & 1 \\ \alpha^{(b-k+1)(n-1)} & \alpha^{(b-k+1)(n-2)} & \cdots & \alpha^{(b-k+1)} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b-1)(n-1)} & \alpha^{(b-1)(n-2)} & \cdots & \alpha^{(b-1)} & 1 \\ \alpha^{b(n-1)} & \alpha^{b(n-2)} & \cdots & \alpha^{b} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b+\delta-1)(n-1)} & \alpha^{(b+\delta-1)(n-2)} & \cdots & \alpha^{(b+\delta-1)} & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \\ c_k \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{k-1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

The evaluations m_i, $i \in [0, \ldots, k-1]$ can be any element from $\mathbb{F}_{q^m}$ (not necessarily equal to zero) since they are non-roots. Isolating the lower part of the matrix equation for which evaluations are required to be zero i.e. evaluations at the roots of

$c(x)$,

$$\begin{bmatrix} \alpha^{b(n-1)} & \alpha^{b(n-2)} & \cdots & \alpha^{b} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b+\delta-1)(n-1)} & \alpha^{(b+\delta-1)(n-2)} & \cdots & \alpha^{(b+\delta-1)} & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{k-1} \\ c_k \\ \vdots \\ c_{n-1} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

Recall that the parity check matrix is a matrix $\mathbf{H}$ such that,

$$\mathbf{c}\mathbf{H}^T = \mathbf{c}^T\mathbf{H} = \mathbf{0}$$

The parity check matrix of the RS code is then,

$$\mathbf{H} = \begin{bmatrix} \alpha^{b(n-1)} & \alpha^{b(n-2)} & \cdots & \alpha^{b} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b+\delta-1)(n-1)} & \alpha^{(b+\delta-1)(n-2)} & \cdots & \alpha^{(b+\delta-1)} & 1 \end{bmatrix}$$

which is a Vandermonde matrix.

3.3 BCH Codes

An RS code has a set of cyclically consecutive roots $V = \{\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-1}\}$ with cardinality δ. A subfield subcode of this RS code is a BCH code restricted to $\mathbb{F}_q$. A BCH code consists of codewords of an RS code that have symbols only in $\mathbb{F}_q$ and as a consequence, in addition to the consecutive roots of the RS code the BCH code will have additional roots that are co-members with the consecutive roots in their respective conjugacy classes. Recall the definition of a conjugacy class from Definition 2.3. The set of roots of a BCH code are given by

$$R = \bigcup_{\beta \in V} C(\beta), \tag{3.2}$$

the codes have redundancy $|R|$ and dimension $k = n - |R|$. Clearly $V \subset R$ and the minimum distance of BCH codes is at least $|V| + 1 = r + 1$. Often R contains one or more roots that are cyclically consecutive to the set of roots in V. If $T \subset R$ denotes this additional set of consecutive roots with $T \nsubseteq V$ then the minimum distance of the BCH code is

$$d \geq |V| + |T| + 1$$

from the BCH bound. As a result of restricting an RS to a subfield we obtain a BCH code with a reduced dimension, the same length and often the same minimum distance as the original RS code.

Example 3.1: Consider the RS code defined with the roots $V = \{\alpha^0, \alpha^1, \alpha^2\}$ in $\mathbb{F}_{16}$. This code has parameters $[15, 12, 4]_{16}$. The parity check matrix of the RS code is given by,

$$\mathbf{H}_{rs} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 \\ \alpha^{14} & \alpha^{13} & \cdots & \alpha^2 & \alpha & 1 \\ \alpha^{13} & \alpha^{11} & \cdots & \alpha^4 & \alpha^2 & 1 \end{bmatrix}.$$

The conjugacy classes of $\mathbb{F}_{16}$ over $\mathbb{F}_2$ are

$$\begin{aligned} \{\alpha^0\} \\ \{\alpha, \alpha^2, \alpha^4, \alpha^8\} \\ \{\alpha^3, \alpha^6, \alpha^{12}, \alpha^9\} \\ \{\alpha^5, \alpha^{10}\} \\ \{\alpha^7, \alpha^{14}, \alpha^{13}, \alpha^{11}\} \end{aligned}$$

The set of the defining roots of the BCH code in $\mathbb{F}_2$ is $R = \{\alpha^0, \alpha, \alpha^2, \alpha^4, \alpha^8\}$ and has three cyclically consecutive roots. The parity check of the BCH code is given by,

$$\mathbf{H}_{bch} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 1 \\ \alpha^{14} & \alpha^{13} & \cdots & \alpha^2 & \alpha & 1 \\ \alpha^{13} & \alpha^{11} & \cdots & \alpha^4 & \alpha^2 & 1 \\ \alpha^{13} & \alpha^{7} & \cdots & \alpha^8 & \alpha^4 & 1 \\ \alpha^{7} & \alpha^{14} & \cdots & \alpha & \alpha^8 & 1 \end{bmatrix}.$$

The BCH code therefore has parameters $[15, 10, 4]_2$. To obtain the best possible dimension it is desirable that the BCH code defined by (11.19) to be *narrow sense*. Narrow sense BCH codes have defining roots in $\mathbb{F}_{q^m}$

$$V = \{\alpha, \alpha^2, \ldots, \alpha^{\delta-1}\}$$

Narrow sense BCH codes tend to have the cardinality $|R|$ to be comparatively small when $|A| = r_1$ is also small. A BCH code is said to be *primitive* if it has length $n = |\mathbb{F}_{q^m}| - 1$.

3.4 Goppa Codes

Goppa (1970, 1971) introduced a class of linear codes commonly referred to as *Goppa codes* or $\Gamma(L,G)$ codes. Goppa codes meet the well known Gilbert-Varshamov bound (see Section 4.2). $\Gamma(L,G)$ codes have good properties and some of these codes have the best known minimum distance of any known codes with the same length and rate. Goppa codes are also used extensively in cryptography in public key cryptosystems. The codes are mainly defined in a finite field $\mathbb{F}_q$ and are subfield subcodes of generalised Reed Solomon codes defined in an extension field of $\mathbb{F}_q$.
A $\Gamma(L,G)$ code is defined by a set $L \subseteq \mathbb{F}_{q^m}$ and a polynomial $G(x)$ with coefficients from $\mathbb{F}_{q^m}$, where $\mathbb{F}_{q^m}$ is a finite extension of the field $\mathbb{F}_q$. The set $L = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ with cardinality n contains all elements of $\mathbb{F}_{q^m}$ that are not roots of the Goppa polynomial $G(x)$. A codeword $(c_0, c_1, \ldots c_{n-1})$ with elements from $\mathbb{F}_q$ is a word of a Goppa code defined by the set L and the polynomial $G(x)$ if it satisfies

$$\sum_{i=0}^{n-1} \frac{c_i}{x-\alpha_i} \equiv 0 \bmod G(x). \tag{3.3}$$

If r is the degree of the polynomial $G(x) \in \mathbb{F}_{q^m}[x]$ the parameters of the Goppa code are:

$$\begin{array}{ll} \text{length:} & n = |L|, \\ \text{redundancy:} & n-k \leq mr, \\ \text{distance:} & d \geq r+1. \end{array}$$

From (MacWilliams and Sloane, 1983) we know that in the modulo ring $\mathbb{F}_{q^m}[x]/G(x)$, $(x-\alpha_i)$ has a inverse since it does not divide $G(x)$.

$$(x-\alpha_i)^{-1} = -\frac{G(x)-G(\alpha_i)}{x-\alpha_i} G(\alpha_i)^{-1} \tag{3.4}$$

Substituting (3.4) in (3.3) we have,

$$\sum_{i=0}^{n-1} c_i \frac{G(x)-G(\alpha_i)}{x-\alpha_i} G(\alpha_i)^{-1} = 0$$

Let $G(x) = \sum_{i=0}^{r} g_i x^i$ and $g_r \neq 0$ then

$$\begin{aligned} \frac{G(x)-G(\alpha_i)}{x-\alpha_i} =& g_r(x^{r-1} + x^{r-2}\alpha_i + \cdots + \alpha_i^{r-1}) + \\ & g_{r-1}(x^{r-2} + x^{r-3}\alpha_i + \cdots + \alpha_i^{r-2}) + \cdots + \\ & g_2(x+\alpha_i) + \\ & g_1 \end{aligned} \tag{3.5}$$

Equating the coefficients of $x^{r-1}, x^{r-2}, \ldots, 1$ to zero in (3.5) we have the matrix H,

$$H = \begin{bmatrix} \frac{g_r}{G(\alpha_0)} & \cdots & \frac{g_r}{G(\alpha_{n-1})} \\ \frac{g_{r-1}+\alpha_0 g_r}{G(\alpha_0)} & \cdots & \frac{g_{r-1}+\alpha_{n-1} g_r}{G(\alpha_{n-1})} \\ \vdots & \ddots & \vdots \\ \frac{g_1+\alpha_0 g_2+\cdots+\alpha_0^{r-1} g_r}{G(\alpha_0)} & \cdots & \frac{g_1+\alpha_{n-1} g_2+\cdots+\alpha_{n-1}^{r-1} g_r}{G(\alpha_{n-1})} \end{bmatrix}$$

for which $c = (c_0, \ldots, c_{n-1})$, $cH^{\mathrm{T}} = 0$. The matrix H can be expanded such that,

$$H = \begin{bmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_r \end{bmatrix} \begin{bmatrix} \frac{1}{G(\alpha_0)} & \frac{1}{G(\alpha_1)} & \cdots & \frac{1}{G(\alpha_{n-1})} \\ \frac{\alpha_0}{G(\alpha_0)} & \frac{\alpha_1}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}}{G(\alpha_{n-1})} \\ \frac{\alpha_0^2}{G(\alpha_0)} & \frac{\alpha_1^2}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^2}{G(\alpha_{n-1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_0^{r-1}}{G(\alpha_0)} & \frac{\alpha_1^{r-1}}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^{r-1}}{G(\alpha_{n-1})} \end{bmatrix}.$$

The matrix

$$\begin{bmatrix} g_r & 0 & 0 & \cdots & 0 \\ g_{r-1} & g_r & 0 & \cdots & 0 \\ g_{r-2} & g_{r-1} & g_r & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_1 & g_2 & g_3 & \cdots & g_r \end{bmatrix}$$

is invertible and need not be used, as a result the parity check matrix of a Goppa code is given by

$$H = \begin{bmatrix} \frac{1}{G(\alpha_0)} & \frac{1}{G(\alpha_1)} & \cdots & \frac{1}{G(\alpha_{n-1})} \\ \frac{\alpha_0}{G(\alpha_0)} & \frac{\alpha_1}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}}{G(\alpha_{n-1})} \\ \frac{\alpha_0^2}{G(\alpha_0)} & \frac{\alpha_1^2}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^2}{G(\alpha_{n-1})} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_0^{r-1}}{G(\alpha_0)} & \frac{\alpha_1^{r-1}}{G(\alpha_1)} & \cdots & \frac{\alpha_{n-1}^{r-1}}{G(\alpha_{n-1})} \end{bmatrix}.$$

Goppa codes are a large class and include the well known BCH codes as a sub-class.

3.2 *Theorem (From* (MacWilliams and Sloane, 1983)***).*** *A $\Gamma(L,G)$ defined with the polynomial $G(x) = x^r$ and the set $L = \{\mathbb{F}_{q^m} \setminus \{0\}\}$ corresponds to a BCH code defined in $\mathbb{F}_q$ with length $n = q^m - 1$.*

Separable Goppa codes are $\Gamma(L,G)$ defined by a square-free polynomial $G(x)$, i.e., $G(x)$ has distinct roots each having multiplicity exactly one. An *irreducible* Goppa code is one which is defined by a Goppa polynomial which is irreducible over its coefficient field $\mathbb{F}_{q^m}$.
Goppa codes defined in $\mathbb{F}_{q^m}$ are GRS codes. The definition of generalised Reed Solomon (GRS) codes is recalled from (MacWilliams and Sloane, 1983). A GRS code, denoted by $\mathrm{GRS}_k(\alpha,\mathbf{v})$, consists of all the vectors,

$$(v_1F(\alpha_1),v_2F(\alpha_2),\ldots,v_nF(\alpha_n)))$$

where $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_n)$ consists of distinct elements of $\mathbb{F}_{q^m}$, a template $\mathbf{v}=(v_1,v_2,\ldots,v_n)$ consists of arbitrary elements from $\mathbb{F}_{q^m}$ none of which is zero and $F(x)$ is a polynomial of degree at most $k-1$. Also from (MacWilliams and Sloane, 1983) it is shown that Goppa codes defined by some $G(x)$ of degree r and the set $L=\{\alpha_1,\alpha_2,\ldots,\alpha_n\}$ are sub-field sub-codes of $\mathrm{GRS}_{n-r}(\alpha,\mathbf{v})$ with $k=n-r$ and,

$$v_i=\frac{G(\alpha_i)}{\prod_{j\neq i}(\alpha_i-\alpha_j)},\quad i=1,\ldots,n. \tag{3.6}$$

Again from (MacWilliams and Sloane, 1983) observe that dual code of a $\mathrm{GRS}_k(\alpha,\mathbf{v})$ code is also a GRS code of the form $\mathrm{GRS}_{n-k}(\alpha,\acute{\mathbf{v}})$ for some template $\acute{\mathbf{v}}$.

Binary Goppa Codes Binary Goppa codes are the most studied type of Goppa codes. Some distance properties of binary Goppa codes are given. If a Goppa polynomial $G(x)$ has distinct non-repeated roots then the resulting binary Goppa code has distance $d\geq 2\deg(G(x))+1$. If a Goppa polynomial $G(x)$ has repeated roots then the minimum distance of the resulting binary Goppa code is $d\geq\deg(\acute{G}(x))+1$ where $\acute{G}(x)$ is the smallest degree perfect square polynomial that is divisible by $G(x)$. If a Goppa polynomial $G(x)=G_1(x)G_2(x)$ with $G_1(x)$ having distinct non-repeated roots and $G_2(x)$ has repeated roots then the resulting binary Goppa code has minimum distance $d\geq 2\deg(G_1(x))+\deg(\acute{G}_2(x))+1$ (MacWilliams and Sloane, 1983, Ch. 12).

3.5 Summary

Three important classes of algebraic codes have been discussed. These codes are perhaps the most studied codes in coding theory and the short introduction in this chapter only aims to serve the purposes of subsequent chapters. In practice BCH and RS codes have found extensive use in communication systems whereas Goppa codes are used in cryptosystems. These codes still an active area of research especially with the introduction of list decoding by Guruswami and Sudan (1999) that extended the error correction capability of RS and BCH codes. Chapter 4 introduces algebraic geometry (AG) codes which are evaluations of multivariate functions on

a curve in a 2-dimensional plane. A more detailed treatment of these codes can be found in (Blahut, 2008; MacWilliams and Sloane, 1983; Shu and Costello, 2004).

4. Algebraic Geometry Codes

4.1 Introduction

In order to meet channel capacity, as Shannon postulated, long error correction codes with large minimum distances need to be found. A large effort in research has been dedicated to finding algebraic codes with good properties and efficient decoding algorithms. Reed Solomon (RS) codes are a product of this research and have over the years found numerous applications, the most noteworthy being their implementation in satellite systems and compact discs. These codes are defined with non-binary alphabets and have the maximum achievable minimum distance for codes of their lengths. A generalisation of RS codes was introduced by Goppa using a unique construction of codes from algebraic curves. This development led to active research in that area so that currently the complexity of encoding and decoding these codes has been reduced greatly from when they were first presented. These codes are AG codes and have much greater lengths than RS codes with the same alphabets. Furthermore these codes can be improved if curves with desirable properties can be found. AG codes have good properties and some families of these codes have been shown to be asymptotically superior as they exceed the well-known Gilbert-Varshamov bound (Tsfasman et al., 1982) when the defining finite field $\mathbb{F}_q$ has size $q \geq 49$ with q always a square.

4.2 Bounds Relevant to Algebraic Geometry Codes

Bounds on the performance of codes that are relevant to AG codes are presented in order to show the performance of these codes. Let $A_q(n,d)$ represent the number of codewords in the code space of a code $\mathscr{C}$ with length n, minimum distance d and defined over a field of size q. Let the information rate be $R = k/n$ and the relative minimum distance be $\delta = d/n$ for $0 \leq \delta \leq 1$ then

$$\alpha_q(\delta) = \lim_{n\to\infty} \frac{1}{n} \log(A_q(n, \delta n))$$

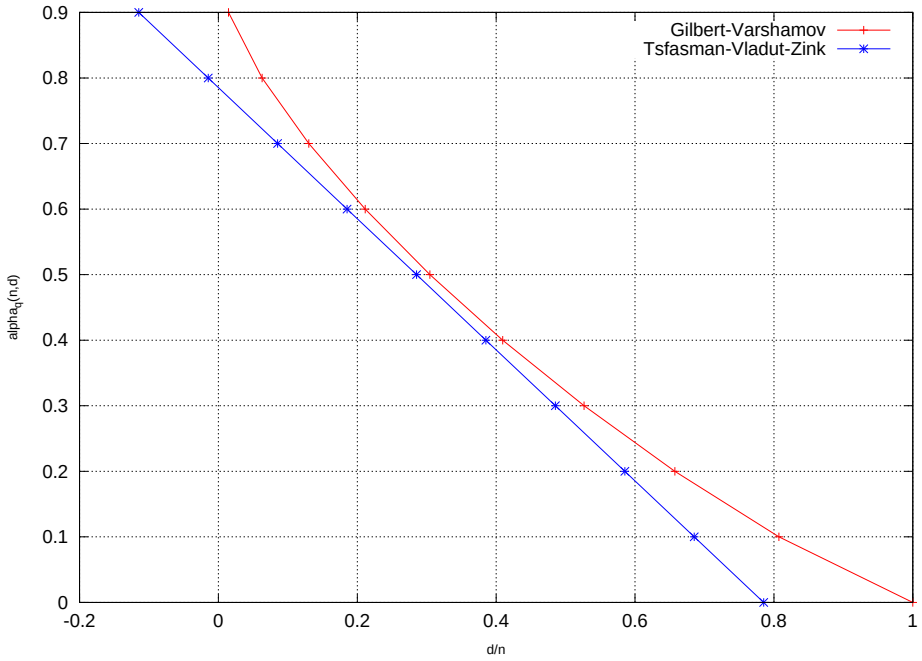

Fig. 4.1: Tsfasman Vladut Zink and Gilbert Varshamov Bound for $q = 32$

which represents the k/n such that there exists a code over a field of size q that has d/n converging to δ (Walker, 2000). The q-ary entropy function is given by

$$H_q(x) = \begin{cases} 0, & x = 0 \\ x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x), & 0 < x \leq \theta \end{cases}$$

The asymptotic Gilbert-Varshamov lower bound on $\alpha_q(\delta)$ is given by,

$$\alpha_q(\delta) \geq 1 - H_q(\delta) \text{ for } 0 \leq \delta \leq \theta$$

The Tsfasman Vladut Zink bound is a lower bound on $\alpha_q(\delta)$ and holds true for certain families of AG codes, it is given by

$$\alpha_q(\delta) \geq 1 - \delta - \frac{1}{\sqrt{q}-1} \text{ where } \sqrt{q} \in \mathbb{N}/0$$

The supremacy of AG codes lies in the fact that the TVZ bound ensures that these codes have better performance when q is a perfect square and $q \geq 49$. The Figures 4.1 to 4.3 show the R vs δ plot of these bounds for some range of q.

4.3 Motivation for Studying AG Codes

Aside from their proven superior asymptotic performance when the field size $q^2 > 49$, AG codes defined in much smaller fields have very good parameters. A closer

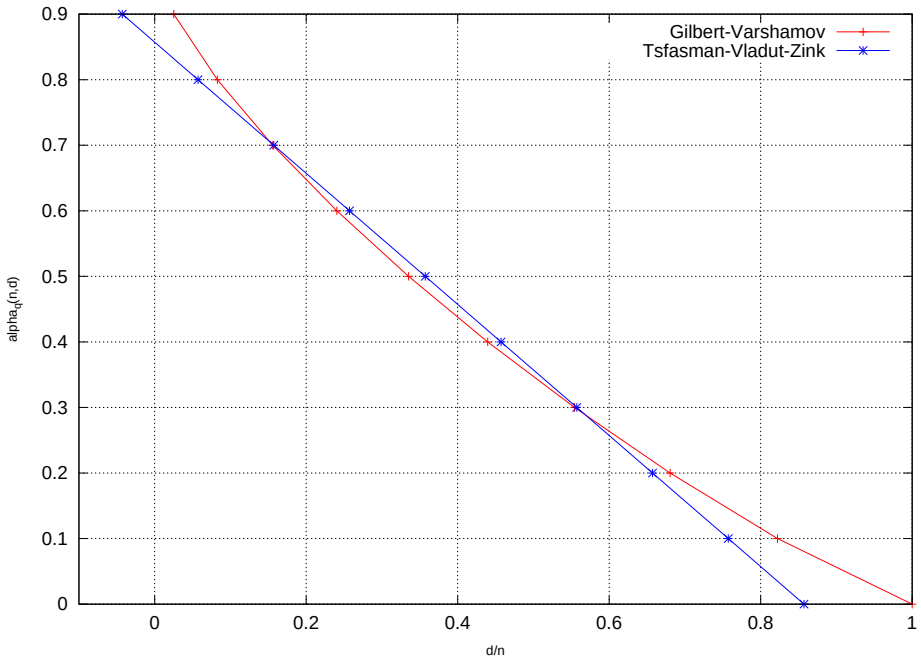

Fig. 4.2: Tsfasman Vladut Zink and Gilbert Varshamov Bound for $q = 64$

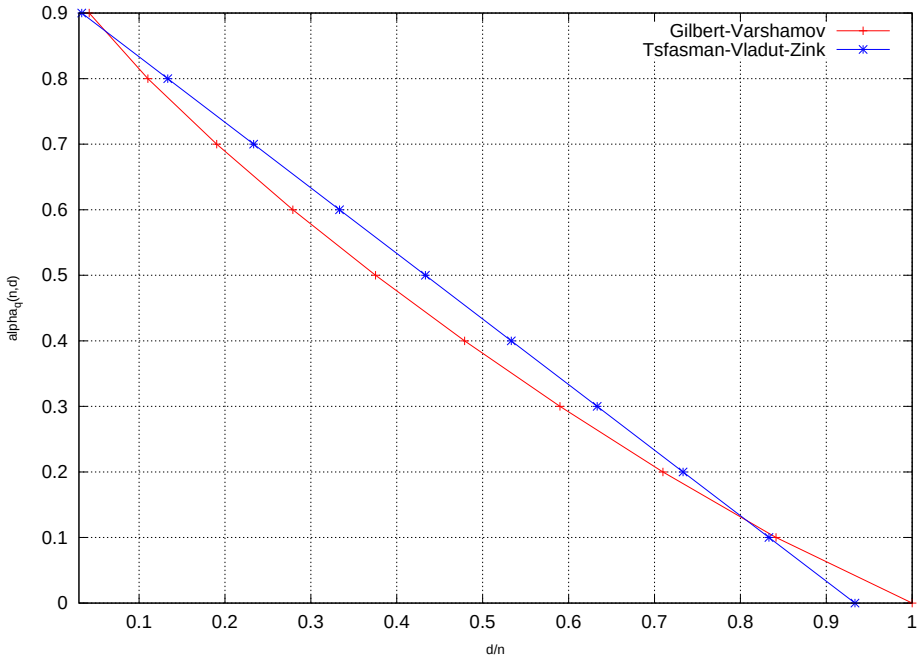

Fig. 4.3: Tsfasman Vladut Zink and Gilbert Varshamov Bound for $q = 256$

look at tables of best known codes in (Grassl, 2007) and (Schimd and Shurer, 2004) shows that algebraic geometry codes feature as the best known linear codes for an appreciable range of code lengths for different field sizes q. To demonstrate a comparison the parameters of AG codes with shortened BCH codes in fields with small sizes and characteristic 2 is given. AG codes of length n, dimension k have minimum distance $d = n-k-g+1$ where g is called the genus . Notice that $n-k+1$ is the distance of a maximum distance (MDS) separable code. The genus g is then the Singleton defect s of an AG code. The Singleton defect is simply the difference between the distance of a code and the distance some hypothetical MDS code of the same length and dimension. Similarly a BCH code is a code with length n, dimension k and distance $d = n-k-s+1$ where s is the Singleton defect and number of non-consecutive roots of the BCH code. Consider Table 4.1 which compares the parameters of AG codes from three curves with genera 3, 7 and 14 with shortened BCH codes with similar code rates. At high rates, BCH codes tend to have better minimum distances or smaller Singleton defects. This is because the roots of the BCH code with high rates are usually cyclically consecutive thus contribute to the minimum distance. For rates close to half AG codes are better than BCH codes since the number of non-consecutive roots of the BCH code is increased as a result of conjugacy classes. The AG codes benefit from the fact that their Singleton defect or genus remains fixed for all rates. As a consequence AG codes significantly outperform BCH codes at lower rates. However the genera of curves with many points in small finite fields are usually large and as the length of the AG codes increases in $\mathbb{F}_8$, the BCH codes beat AG codes in performance. Tables 4.2 and 4.3 show the comparison between AG and BCH codes in fields $\mathbb{F}_{16}$ and $\mathbb{F}_{32}$ respectively. With larger field sizes, curves with many points and small genera can be used and AG codes do much better than BCH codes. It is worth noting that Tables 4.1-4.3 show codes in fields with size less than 49.

Rate	AG code in $\mathbb{F}_{2^3}$	Number of points	Genus	Shortened BCH code in $\mathbb{F}_{2^3}$	BCH code in $\mathbb{F}_{2^3}$
0.2500	[23,5,16]	24	3	[23,5,12]	[63,45,12]
0.3333	[23,7,14]	24	3	[23,7,11]	[63,47,11]
0.5000	[23,11,10]	24	3	[23,10,8]	[63,50,8]
0.6667	[23,15,6]	24	3	[23,14,6]	[63,54,6]
0.7500	[23,17,4]	24	3	[23,16,5]	[63,56,5]
0.8500	[23,19,2]	24	3	[23,18,4]	[63,58,4]
0.2500	[33,8,19]	34	7	[33,7,16]	[63,37,16]
0.3333	[33,11,16]	34	7	[33,11,14]	[63,41,14]
0.5000	[33,16,11]	34	7	[33,15,12]	[63,45,12]
0.6667	[33,22,5]	34	7	[33,22,7]	[63,52,7]
0.7500	[33,24,3]	34	7	[33,24,6]	[63,54,6]
0.2500	[64,16,35]	65	14	[64,16,37]	[63,15,37]
0.3333	[64,21,30]	65	14	[64,20,31]	[63,19,31]
0.5000	[64,32,19]	65	14	[64,31,22]	[63,30,22]
0.6667	[64,42,9]	65	14	[64,42,14]	[63,41,14]
0.7500	[64,48,3]	65	14	[64,48,11]	[63,47,11]

Table 4.1: Comparison between BCH and AG codes in $\mathbb{F}_8$

Rate	AG code in $\mathbb{F}_{2^4}$	Number of points	Genus	Shortened BCH code in $\mathbb{F}_{2^4}$	BCH code in $\mathbb{F}_{2^4}$
0.2500	[23,5,18]	24	1	[23,4,11]	[255,236,11]
0.3333	[23,7,16]	24	1	[23,6,10]	[255,238,10]
0.5000	[23,11,12]	24	1	[23,10,8]	[255,242,8]
0.6667	[23,15,8]	24	1	[23,14,6]	[255,246,6]
0.7500	[23,17,6]	24	1	[23,16,5]	[255,248,5]
0.8500	[23,19,4]	24	1	[23,18,4]	[255,250,4]
0.2500	[64,16,43]	65	6	[64,16,27]	[255,207,27]
0.3333	[64,21,38]	65	6	[64,20,25]	[255,211,25]
0.5000	[64,32,27]	65	6	[64,32,19]	[255,223,19]
0.6667	[64,42,17]	65	6	[64,41,13]	[255,232,13]
0.7500	[64,48,11]	65	6	[64,47,10]	[255,238,10]
0.8500	[64,54,5]	65	6	[64,53,7]	[255,244,7]
0.2500	[126,31,76]	127	20	[126,30,57]	[255,159,57]
0.3333	[126,42,65]	127	20	[126,41,48]	[255,170,48]
0.5000	[126,63,44]	127	20	[126,63,37]	[255,192,37]
0.6667	[126,84,23]	127	20	[126,84,24]	[255,213,24]
0.7500	[126,94,13]	127	20	[126,94,19]	[255,223,19]

Table 4.2: Comparison between BCH and AG codes in $\mathbb{F}_{16}$

4.4 Curves and Planes

In this section the notion of curves and planes is introduced. Definitions and discussions are restricted to two-dimensional planes and all polynomials are assumed to be defined with coefficients in the finite field $\mathbb{F}_q$. The section draws from the following sources (Blake et al., 1998; Massimo, 2003; Van-Lint, 1990; Walker, 2000).A two dimensional affine plane denoted by $\mathbb{A}^2(\mathbb{F}_q)$ is a set of points ,

$$\mathbb{A}^2(\mathbb{F}_q) = \{(\alpha,\beta) : \alpha,\beta \in \mathbb{F}_q\}$$

which has cardinality q^2. Let $f(x,y)$ be a polynomial in the bivariate ring $\mathbb{F}_q[x,y]$.

4.1 *Definition (Curve).* *A curve is the set of points for which the polynomial $f(x,y)$ vanishes to zero. Mathematically, a curve $\mathscr{X}$ is associated with a polynomial $f(x,y)$ so that $f(P) = \{0 | P \in \mathscr{X}\}$.*

A curve $\mathscr{X}$ is called an affine curve if $\mathscr{X} \subset \mathbb{A}^2(\mathbb{F}_q)$. A two dimensional projective plane $\mathbb{P}^2(\mathbb{F}_q)$ is the algebraic closure of $\mathbb{A}^2$ and is defined as the set of equivalence points,

$$\mathbb{P}^2(\mathbb{F}_q) = \{(\alpha:\beta:1) : \alpha,\beta \in \mathbb{F}_q\} \cup \{(\alpha:1:0) : \alpha \in \mathbb{F}_q\} \cup \{(1:0:0)\}.$$

A curve $\mathscr{X}$ is said to lie in the projective plane if $\mathscr{X} \subset \mathbb{P}^2(\mathbb{F}_q)$. The affine polynomial $f(x,y)$ is in two variables, in order to define a projective polynomial in three

Rate	AG code in $\mathbb{F}_{2^4}$	Number of points	Genus	Shortened BCH code in $\mathbb{F}_{2^4}$	BCH code in $\mathbb{F}_{2^4}$
0.2500	[43,10,33]	44	1	[43,10,18]	[1023,990,18]
0.3333	[43,14,29]	44	1	[43,14,16]	[1023,994,16]
0.5000	[43,21,22]	44	1	[43,20,13]	[1023,1000,13]
0.6667	[43,28,15]	44	1	[43,28,9]	[1023,1008,9]
0.7500	[43,32,11]	44	1	[43,32,7]	[1023,1012,7]
0.8500	[43,36,7]	44	1	[43,36,5]	[1023,1016,5]
0.2500	[75,18,53]	76	5	[75,18,30]	[1023,966,30]
0.3333	[75,25,46]	76	5	[75,24,27]	[1023,972,27]
0.5000	[75,37,34]	76	5	[75,36,21]	[1023,984,21]
0.6667	[75,50,21]	76	5	[75,50,14]	[1023,998,14]
0.7500	[75,56,15]	76	5	[75,56,11]	[1023,1004,11]
0.8500	[75,63,8]	76	5	[75,62,8]	[1023,1010,8]
0.2500	[103,25,70]	104	9	[103,25,42]	[1023,945,42]
0.3333	[103,34,61]	104	9	[103,33,38]	[1023,953,38]
0.5000	[103,51,44]	104	9	[103,50,28]	[1023,970,28]
0.6667	[103,68,27]	104	9	[103,68,19]	[1023,988,19]
0.7500	[103,77,18]	104	9	[103,76,15]	[1023,996,15]
0.8500	[103,87,8]	104	9	[103,86,10]	[1023,1006,10]

Table 4.3: Comparison between BCH and AG codes in $\mathbb{F}_{32}$

variables *homogenisation* is used,

$$f(x,y,z) = z^d f\left(\frac{x}{z}, \frac{y}{z}\right) \qquad d = \text{Degree of } f(x,y)$$

which turns $f(x,y)$ into a homogeneous[1] polynomial in three variables. The points in the projective plane are called equivalence points since for any point $P \in \mathbb{P}^2(\mathbb{F}_q)$,

$$\text{if } f(x_0,y_0,z_0) = 0 \quad \text{then } f(\alpha x_0, \alpha y_0, \alpha z_0) = 0 \qquad \alpha \in \mathbb{F}_q^*,\ P = (x_0 : y_0 : z_0)$$

because $f(x,y,z)$ is homogeneous. The colons in the notation of a projective point $(x:y:z)$ represents this equivalence property. The affine space $\mathbb{A}^2(\mathbb{F}_q)$ is a subset of $\mathbb{P}^2(\mathbb{F}_q)$ and is given by,

$$\mathbb{A}^2(\mathbb{F}_q) = \{(\alpha : \beta : 1) : \alpha, \beta \in \mathbb{F}_q\} \subset \mathbb{P}^2(\mathbb{F}_q).$$

A projective curve can then be defined as a set of points,

$$\mathscr{X} = \{P : f(P) = 0, P \in \mathbb{P}^2(\mathbb{F}_q)\}.$$

A point on a projective curve $\mathscr{X}$ that coincides with any of the points of $\mathbb{P}^2(\mathbb{F}_q)$ of the form,

$$\{(\alpha : 1 : 0) : \alpha \in \mathbb{F}_q\} \cup \{(1 : 0 : 0)\}$$

i.e. points $(x_0 : y_0 : z_0)$ for which $z_0 = 0$ is called a point at infinity. A third plane called the bicyclic plane (Blahut, 2008) is a subset of the $\mathbb{A}^2(\mathbb{F}_q)$ and consists of

[1] Each term in the polynomial has degree equal to d.

points,

$$\{(\alpha,\beta) : \alpha,\beta \in \mathbb{F}_q \setminus \{0\}\}.$$

This plane was defined so as to adapt the Fourier transform to AG codes since the inverse Fourier transform is undefined for zero coordinates. A curve associated with a polynomial $f(x,y,z)$ that cannot be reduced or factorised is called *irreducible*. A point on a curve is singular if its evaluation on all partial derivatives of the defining polynomial with respect to each indeterminate is zero. Suppose f_x, f_y and f_z denote partial derivatives of $f(x,y,z)$ with respect to x, y and z respectively. A point $P \in \mathscr{X}$ is singular if,

$$f_x(P) = f_y(P) = f_z(P) = 0.$$

A curve $\mathscr{X}$ is nonsingular or smooth does not contain any singular points. To obtain AG codes it is required that the defining curve is both irreducible and smooth. The genus of a curve can be seen as a measure of how many bends a curve has on its plane. The genus of a smooth curve defined by $f(x,y,z)$ is given by the Plücker formula,

$$g = \frac{(d-1)(d-2)}{2} \quad d = \text{Degree of } f(x,y,z)$$

The genus plays an important role in determining the quality of AG codes. It is desirable for curves that define AG codes to have small genera.

Example 4.1: Consider the Hermitian curve in $\mathbb{F}_4$ defined as,

$$f(x,y) = x^3 + y^2 + y \quad \text{affine}$$
$$f(x,y,z) = x^3 + y^2 z + y z^2 \quad \text{projective}$$

It is straightforward to verify that the curve is irreducible. The curve has the following projective points,

$$(0:0:1) \quad (0:1:1) \quad (\alpha:\alpha:1) \quad (\alpha:\alpha^2:1)$$
$$(\alpha^2:\alpha:1) \quad (\alpha^2:\alpha^2:1) \quad (1:\alpha:1) \quad (1:\alpha^2:1) \quad (0:1:0)$$

Notice the curve has a single point at infinity $P_\infty = (0:1:0)$. One can easily check that the curve has no singular points and is thus smooth.

4.5 Important Theorems and Concepts

The length of an AG code is equal to the number of points on the defining curve. Since it is desirable to obtain codes that are as long as possible, it is desirable to

know what the maximum number of points attainable from a curve given a genus is.

4.1 Theorem (Hasse Weil with Serre's Improvement (Blake et al., 1998)). *The Hasse Weil theorem with Serre's improvement says that the number of rational points[2] of an irreducible curve, n, with genus g in $\mathbb{F}_q$ is upper bounded by,*

$$n \leq q + 1 + g\lfloor 2\sqrt{q} \rfloor.$$

Curves that meet this bound are called *maximal* curves. The Hermitian curves are examples of maximal curves. Bezout's theorem is an important theorem and is used to determine the minimum distance of algebraic geometry codes. It describes the size of the set which is the intersection of two curves in the projective plane.

4.2 Theorem (Bezout's Theorem (Blake et al., 1998)). *Any two curves $\mathcal{X}_a$ and $\mathcal{X}_b$ with degrees of their associated polynomials as m and n respectively, have at most mn common roots in the projective plane counted with multiplicity.*

4.2 Definition (Divisor). *A divisor on a curve $\mathcal{X}$ is a formal sum associated with the points of the curve.*

$$D = \sum_{P \in \mathcal{X}} n_p P$$

where $n_p \geq 0$ are integers.

A zero divisor is one that has $n_p = 0$ for all $P \in \mathcal{X}$. A divisor is called effective if it is not a zero divisor. The support of a divisor is a subset of $\mathcal{X}$ for which $n_p \neq 0$. The degree of a divisor is given as,

$$deg(D) = \sum_{P \in \mathcal{X}} n_p \; deg(P)$$

For simplicity it is assumed that the degree of points $P \in \mathcal{X}$ i.e. $deg(P)$ is 1 (points of higher degree are discussed in Chapter 7). Addition of two divisors $D_1 = \sum_{P \in \mathcal{X}} n_p P$ and $D_2 = \sum_{P \in \mathcal{X}} \acute{n}_p P$ is so defined,

$$D_1 + D_2 = \sum_{P \in \mathcal{X}} (n_p + \acute{n}_p) P.$$

[2] A rational point is a point of degree one. See Chapter 7 for the definition of the degree of point on a curve.

Divisors are simply book keeping structures that store information on points of a curve. Below is an example the intersection divisor of two curves.

Example 4.2: Consider the Hermitian curve in $\mathbb{F}_4$ defined as,

$$f_1(x,y,z) = x^3 + y^2z + yz^2$$

and the curve defined by

$$f_2(x,y,z) = x$$

with points

$$(0:0:1) \quad (0:1:1) \quad (0:\alpha:1) \quad (0:\alpha^2:1) \quad (0:1:0)$$

These two curves intersect at points all with multiplicity 1,

$$(0:0:1) \quad (0:1:0) \quad (0:1:1).$$

Alternatively, this may be represented using a divisor D,

$$D = (0:0:1) + (0:1:0) + (0:1:1)$$

with n_p the multiplicity, equal to 1 for all the points. Notice that the two curves meet at exactly $deg(f_1)deg(f_2) = 3$ points in agreement with Bezout's theorem.

For rational functions with denominators, points in divisor with $n_p < 0$ are poles. For example $D = P_1 - 2P_2$ will denote an intersection divisor of two curves that have one zero P_1 and pole P_2 with multiplicity two in common. Below is the formal definition of the field of fractions of a curve $\mathcal{X}$.

4.3 *Definition (Field of fractions).* *The field of fractions $\mathbb{F}_q(\mathcal{X})$ of a curve $\mathcal{X}$ defined by a polynomial $f(x,y,z)$ contains all rational functions of the form*

$$\frac{g(x,y,z)}{h(x,y,z)}$$

with the restriction that $g(x,y,z)$ and $h(x,y,z)$ are homogeneous polynomials, have the same degree and are not divisible by $f(x,y,z)$.

Elements of a subset (Riemann-Roch space) of the field of fractions of $\mathcal{X}$ meeting certain conditions are evaluated at points of the curve $\mathcal{X}$ to form codewords of an AG code. Thus there is a one-to-one mapping between rational functions in this subset and codewords of an AG code. The Riemann-Roch theorem defines this subset and gives a lower bound on the dimension of AG codes. The definition of a Riemann-Roch space is given.

4.4 Definition (Riemann Roch Space). *The Riemann Roch space associated with a divisor D is given by,*

$$L(D) = \{t \in \mathbb{F}_q(\mathscr{X}) | (t) + D \geq 0\} \cup 0$$

where $\mathbb{F}_q(\mathscr{X})$ is the field of fractions and (t) is the intersection divisor[3] of the rational function t and the curve $\mathscr{X}$.

Essentially the Riemann-Roch space associated with a divisor D is a set of functions t from $\mathbb{F}_q(\mathscr{X})$ such that $(t)+D$ has no poles. The rational functions in $L(D)$ are functions from the field of fractions $\mathbb{F}_q(\mathscr{X})$ that must have poles only in the zeros (positive terms) contained in the divisor D, each pole occurring with at most the multiplicity defined in the divisor D and most have zeros only in the poles (negative terms) contained in the divisor D, each zero occurring with at most the multiplicity defined in the divisor D .

4.3 Theorem (Riemann Roch Theorem (Blake et al., 1998)). *Let $\mathscr{X}$ be a curve with genus g and D any divisor with degree $(D) > 2g-2$, then the dimension of the Riemann Roch space associated with D, denoted by $l(D)$ is,*

$$l(D) = degree(D) - g + 1$$

Algebraic geometry codes are the image of an evaluation map of a Riemann Roch space associated with a divisor D so that

$$L(D) \rightarrow \mathbb{F}_q^n$$

$$t \rightarrow (t(P_1), t(P_2), \ldots, t(P_n))$$

where $\mathscr{X} = \{P_1, P_2, \ldots, P_n, P_x\}$ is a smooth irreducible projective curve of genus g defined over $\mathbb{F}_q$. The divisor D must have no points in common with a divisor T associated with $\mathscr{X}$ i.e. it has support disjoint from T. For example if the divisor T is of the form

$$T = P_1 + P_2 + \cdots + P_n$$

then $D = mP_x$. Codes defined by the divisors T and $D = mP_x$ are called one point AG codes (since the divisor D has a support containing only one point) and AG codes are predominantly defined as so since the parameters of such codes are easily determined (Lachaud et al., 1995).

[3] An intersection divisor is a divisor that contains information on the points of intersection of two curves.

4.6 Construction of AG Codes

The following steps are necessary in order to construct a generator matrix of an AG code,

1. Find the points of a smooth irreducible curve and its genus.
2. Choose divisors D and $T = P_1 + \cdots + P_n$. From the Riemann-Roch theorem determine the dimension of the Riemann-Roch space $L(D)$ associated with divisor D. This dimension $l(D)$ is the dimension of the AG code.
3. Find $k = l(D)$ linearly independent rational functions from $L(D)$. These form the basis functions of $L(D)$.
4. Evaluate all k basis functions on the points in the support of T to form the k rows of a generator matrix of the AG code.

Example 4.3: Consider again the Hermitian curve defined in $\mathbb{F}_4$ as,

$$f(x,y,z) = x^3 + y^2 z + y z^2$$

1. In Example 4.1 this curve was shown to have 8 affine points and one point at infinity. The genus of this curve is given by the Plücker formula,

   $$g = \frac{(r-1)(r-2)}{2} = 1$$

   where $r = 3$ is the degree of $f(x,y,z)$.

2. Let $D = 5P_\infty$ where $P_\infty = (0:1:0)$ and T be the sum of all 8 affine points. The dimension of the Riemann-Roch space is then given by,

   $$l(5P_\infty) = 5 - 1 + 1 = 5$$

   thus the AG code has dimension $k = 5$.

3. The basis functions for the space $L(5P_\infty)$ are

   $$\{t_1, \ldots, t_k\} = \left\{1, \frac{x}{z}, \frac{x^2}{z^2}, \frac{y}{z}, \frac{xy}{z^2}\right\}$$

   By examining the basis it is clear that $t_1 = 1$ has no poles thus $(t_1) + D$ has no poles also. Basis functions with denominator z have $(t_i) = S - P_\infty$ where S is a divisor of the numerator. Thus $(t_i) + D$ has no poles. Basis functions with denominator z^2 have $(t_j) = S - 2P_\infty$ where S is a divisor of the numerator. Thus $(t_j) + D$ also has no poles.

4. The generator matrix of the Hermitian code defined with divisor $D = 5P_\infty$ is thus,

$$G = \begin{bmatrix} t_1(P_1) & \cdots & t_1(P_n) \\ \vdots & \ddots & \vdots \\ t_k(P_1) & \cdots & t_k(P_n) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha^2 & 1 \\ 0 & 1 & 0 & 0 & 0 & \alpha^2 & \alpha & 0 \\ 0 & 0 & 1 & 0 & 0 & \alpha & 1 & \alpha \\ 0 & 0 & 0 & 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

4.4 Theorem (From (Blake et al., 1998)). *The minimum distance of an AG code is given by,*

$$d \geq n - degree(D)$$

Thus the Hermitian code defined by $D = 5P_\infty$ is a $[8,5,3]_4$ code. The dual of an AG code has parameters (Hoholdt et al., 1998),

$$\text{Dimension, } k^\perp = n - degree(D) + g - 1$$
$$\text{Distance, } d^\perp \geq degree(D) - 2g + 2$$

4.6.1 Affine Hermitian and Reed Solomon Codes

In (Justesen et al., 1989) a description of AG codes on the affine plane was given and the construction of these codes does not require deep knowledge of algebraic geometry or use of divisors. Let V_J be a vector space of all homogeneous polynomials of degree at most J with $J < q$ in a finite field $\mathbb{F}_q$. Let the affine points of a nonsingular irreducible curve be the set $\mathcal{X} = \{P_1, P_2, \ldots, P_n\}$ with n points. A Hermitian code $G(J)$ is given as the result of the evaluation,

$$G(J) = \{(f(P_1), f(P_2), f(P_3).....f(P_n)) | f \in V_J\}$$
$$H(J) = G(J)^\perp$$

A polynomial basis of V_J is given as the first k monomials,

$$x^i y^j \quad i \geq 0, j \geq 0. \tag{4.1}$$

If the first k monomials in (4.1) are given as $\{f_1(x,y), f_2(x,y), \ldots, f_k(x,y)\}$ the generator matrix for the code $G(J)$ is given by,

$$G = \begin{bmatrix} f_1(P_1) & f_1(P_2) & \ldots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \ldots & f_2(P_n) \\ . & . & . & . \\ . & . & . & . \\ f_k(P_1) & f_k(P_2) & \ldots & f_k(P_n) \end{bmatrix} \tag{4.2}$$

The dimension k of the code is not arbitrary and is determined by the Riemann Roch theorem,

$$k = mJ - g + 1$$

for a curve of degree m and genus g. For each basis in (4.1), linear dependence between the rows of G is avoided by a careful choice of the monomials. For example the Hermitian code defined by $x^m + y^{m-1} + y = 0$ will have three rows of G corresponding monomials to x^m,y^{m-1} and y linearly dependent. To avoid this the degree of x is restricted so that the monomial basis of the Hermitian code becomes,

$$x^i y^j \quad 0 \leq i < m, j \geq 0. \tag{4.3}$$

Additionally, the monomials are ordered using a graded lexicographic ordering (see Section 5.2.1) so that they are of the form,

$$1, x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3, \ldots\ldots$$

The *designed* minimum distance of the code of length n is determined by Bezout's theorem and is given by

$$d = n - mJ$$

The dual code has the matrix in (4.2) as a parity check matrix. The dimension of V_J for these codes is,

$$k^{\perp} = n - (mJ - g + 1)$$

and minimum distance,

$$d^{\perp} = mJ - 2g + 2$$

The important parameter J is only defined in the range,

$$m - 2 \leq J \leq \left\lfloor \frac{n-1}{m} \right\rfloor \tag{4.4}$$

Consider the code defined by the Hermitian curve in $\mathbb{F}_4$. The finite field is defined with the primitive polynomial $t^2 = t + 1$ and primitive element α. The Hermitian

curve is given as,

$$x^m + y^{m-1} + y = 0$$

where $m = \sqrt{q} + 1 = 3$. The curve has $n = q^{\frac{3}{2}} = 8$ affine points,

$$\begin{array}{cccc} (0,0) & (0,1) & (\alpha,\alpha) & (\alpha,\alpha^2) \\ (\alpha^2,\alpha) & (\alpha^2,\alpha^2) & (1,\alpha) & (1,\alpha^2) \end{array}$$

The curve has genus $g = 1/2(m-1)(m-2) = 1$ and from (4.4), $1 \le J \le 2$. If $J = 2$ according to (4.3) the first $k = mJ - g + 1 = 6$ monomials form a basis,

$$1, x, y, x^2, xy, y^2$$

The generator matrix for $G(J)$ is

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \alpha & \alpha & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 \\ 0 & 0 & 1 & 1 & \alpha^2 & \alpha^2 & \alpha & \alpha \\ 0 & 0 & \alpha & \alpha^2 & \alpha^2 & 1 & 1 & \alpha \\ 0 & 1 & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha \end{bmatrix}$$

The code has minimum distance $d_{min} = 2$ and the dual code $H(J)$ has $H = G$ with $d_{min} = 6$. Reed Solomon codes can be represented using a similar construction. Using the polynomial

$$x + y = 0$$

the curve associated with it $\mathcal{X} = \{(0,0),(1,1),(\alpha,\alpha),(\alpha^2,\alpha^2),\ldots,(\alpha^{q-2},\alpha^{q-2})\}$ is simply a diagonal line with q points in $\mathbb{F}_q$ and a zero genus. Choosing any J in the range $0 \le J \le q-2$ so that,

$$k = q - (J+1)$$

and

$$d = J + 2$$

A monomial basis of the form,

$$1, x, y, x^2, xy, y^2, \ldots$$

with monomials with degree at most J can also be chosen. Polynomials with the indeterminate x as their evaluation will cause linear dependency on the rows of H (since $x = y$ for all points of the curve) and are therefore neglected. The new basis is,

$$1, y, y^2, y^3, \ldots, y^J$$

The H matrix can then be the evaluation of these monomials with the points of the line. The resulting code $H(J)$ is a singly extended RS code with parameters $(q, q-J-1, J+2)$.

4.7 Summary

Algebraic geometry codes are codes obtained from curves. First the motivation for studying these codes is given. From an asymptotic point of view some families of AG codes have superior performance to the previous best known bound on the performance of linear codes, the Gilbert-Varshamov bound. For codes of moderate length AG codes have better minimum distances than their main competitors, non-binary BCH codes with the same rate and length defined in the same finite fields. Theorems and definitions as a precursor to AG codes are given. Key theorems are Bezout's and Riemann-Roch. Examples using the well known Hermitian code in a finite field of cardinality 4 are then provided. Finally a simplified affine description of Hermitian and Reed Solomon codes is presented. This chapter introduces concepts on AG codes that will be used in subsequent chapters.

5. Decoding Algebraic Codes

5.1 Introduction

In this chapter decoding of algebraic codes for two common channels; the AWGN channel and the erasure channel is considered. The BMSA (Sakata, 2010) decoder for AG codes for hard decision decoding and the BMA (Berlekamp, 1974) for the hard decision decoding of RS codes in the AWGN channel are introduced and implemented. The ordered reliability (Fossorier and Lin, 1995) decoder for soft decision decoding of linear block codes in the AWGN channel is also presented. An erasure correction algorithm, the in-place algorithm (Cai et al., 2005), for the erasure channel is also given. This chapter presents results on the performance comparison of AG codes to BCH codes in the erasure and AWGN channels using the standard decoding algorithms for these codes.

5.2 Bounded Distance Decoding

Algebraic geometry codes were discovered as a consequence of the search for generalizations of BCH, Reed Solomon and other algebraic codes. As a result research has focused on implementing already known decoding algorithms for Reed Solomon and BCH codes developed by Peterson, and then Berlekamp and Massey on AG codes. RS and BCH codes could correct errors with various algorithms up to their bounded error correction limit of $(d-1)/2$ with polynomial complexity. Success in decoding AG codes up to the bounded distance was limited until the introduction of majority voting of missing syndromes by Feng and Rao (Feng and Rao, 1993). Subsequent decoding procedures built on this notion to extend the correction capability of the codes. Table 5.1 gives the chronological milestones in the decoding of AG codes up to the designed minimum distance.

Two algorithms have since taken the fore on decoding of AG codes namely, the Feng and Rao algorithm which uses linear algebra only to decode AG codes and the Sakata Berlekamp Massey algorithm which is an extension of the Berlekamp Massey algorithm to 2 dimensions. The two algorithms have since been modified so that they both have the same complexity of $O(n^{3-2/(m+1)})$ when the codes are constructed in an m-dimensional affine space (Blahut, 2008)(Feng and Rao, 1993).

Year	Author	Correction	Decoding Algorithm
1989	Justesen et al	$(d-g-1)/2$	Peterson's Decoding (First attempt)
1989	Pellikan	$(d-1)/2$	Algebraic Geometry (Complex and inefficient)
1990	Skorobogatov and Vladut	$(d-g-1)/2$	
1991	Feng and Rao	$(d-1)/2$	Peterson's (Introduced majority voting)
1992	Justesen et al	$(d-g/2-1)/2$	Peterson's decoding
1993	Ehrhad	$(d-1)/2$	Sugiyama's decoding (Only when $d > 6g$)
1995	Sakata and Jensen	$(d-1)/2$	BM decoding (majority voting)

Table 5.1: Developments in decoding AG codes

5.2.1 Berlekamp Massey Sakata Algorithm

The BMSA (Blahut, 2008; Sakata, 1988, 2010; Sakata et al., 1995) is an extension of the well known Berlekamp Massey Algorithm to codes of multiple dimensions. AG codes are two dimensional codes and the BMSA can be used to correct up to $t = \frac{d-1}{2}$ errors. Whereas the Berlekamp Massey algorithm finds the error locator polynomial of a corrupted codeword, the BMSA finds a set of polynomials whose common roots are locations of errors in the corrupted AG codeword. These sets of polynomials are not unique and form the generators of a *locator* ideal for those errors. These polynomials are only unique for a certain syndrome when represented as a reduced Groebner basis i.e. reduced gaussian form in two dimensions.

5.2.1.1 Preliminaries

Firstly some terms used in the algorithm are defined. Discussions are restricted to the two dimensional plane.

. **Ideal** : Recall the definition of an ideal from Definition 3.1. An ideal is *proper* if it is not zero or $\mathbf{F}[x, y]$ and it is *principal* if there one of its elements which every other one is a multiple of.

. **Generator** : A generator of a principal ideal is a member of that ideal which every other member is a multiple of. A generator set of an ideal is a set of polynomials which are members of that ideal that generate all the members of the ideal.

. **Monomial Order**: A monomial order is the ordering of monomials in polynomials. An ordering is called *total* if there is no ambiguity in the ordering. An ordering on monomials is called *partial* if ambiguities exists on the order. The *bidegree* of a bivariate monomial is the degree of its indeterminates.

An example of a partial order is the division order. In a division order in a monomial $m_1(x,y)$ precedes another $m_2(x,y)$ in the order if $m_1(x,y)$ divides $m_2(x,y)$ without a remainder. Formally, for a division order on monomials (denoted by $<_P$) with bidegrees (i_1,j_1) and (i_2,j_2) , (i_1,j_1) is said to be less than (i_2,j_2) in the ordering if $i_1 < i_2$ and $j_1 < j_2$. Clearly this is a partial order since there are instances when the orders of monomials cannot be resolved. An example of a total order, denoted by $<_T$ is the graded lexicographic ordering in which the monomials are first ordered by the sum of their degrees and then by the order of their indeterminates in the list of alphabets. For example the graded lexicographic order in the ring of polynomials $\mathbf{F}[x,y]$ is $(0,0),(1,0),(0,1),(2,0),(1,1),(0,2),(3,0),(2,1)\ldots$. The degree of a bivariate polynomial is the bidegree of its monomial(called the *leading monomial* with the largest degree in a total order. Consequently polynomials can be sorted in a total order by comparing the bidegrees of their leading monomials.

. **Footprint of an ideal**: The footprint or Delta-set of an ideal is set of all bivariate polynomial degrees whose polynomials will divide any member of the ideal without remainders. Alternatively, the footprint of an ideal is a set of all bidegrees that are less in the division order ($<_P$) than any polynomial member of the ideal.

. **Minimal Basis**: A minimal basis is a generator set of an ideal that consists of only monic polynomials (polynomials whose leading coefficient is 1) and whose footprint is the footprint of the ideal.

The BMSA accepts the two dimensional syndrome $S_{(a,b)}$ of an AG code and finds the polynomials whose roots are error locations. The graded lexicographic ordering is used to order the elements of the syndrome array and the locator polynomials. The algorithm proceeds by processing the syndrome array element by element and at each stage the locator polynomials are checked if they meet the recursive relationship at that point of the syndrome array. The recursive relationship is given by,

$$\sum_{\bar{k}}^{\bar{s}_i} F^i_{\bar{k}} S_{\bar{k}+(\bar{r}-\bar{s}_i)} = 0 \quad \text{for all } i \tag{5.1}$$

where S is the syndrome array at a point of bidegree $\bar{r}$ and F^i is the ith polynomial with bidegree $\bar{s}_i$. The set $\mathbf{F} = \{F^0, F^1, \ldots, F^l\}$ consists of polynomials that have satisfied (5.1) at a previous stage in the algorithm. A nonzero value for (5.1) by a polynomial in $\mathbf{F}$ is called a discrepancy and polynomials that produce this discrepancy are updated. Not all polynomials are tested by (5.1), since the recursion holds valid only when $\bar{r} - \bar{s}_i \geq \bar{0}$ polynomials that do not meet this criterion are not

tested and are not updated. An important property of each polynomial in the set **F** is called the span of the polynomial so that,

$$\text{span}(F^i) = \bar{r} - \bar{s}_i.$$

The footprint, $\Delta_{\bar{r}}$, is calculated at each stage based on the bidegrees of the polynomials in **F**. The footprint consists of all bidegrees less than any of the bidegrees of the polynomials in **F** in the partial order $<_P$. Another important set is the set of *interior polynomials* denoted by $\mathbf{G} = \{G^0, G^1, \ldots, G^{l-1}\}$ which contains polynomials that were previously in **F** at some stage in the algorithm with a nonzero discrepancy and whose spans at that stage correspond to the largest bidegrees in the footprint at the present stage. The set **G** is used to update polynomials with a nonzero disrepancy in **F**.

5.2.1.2 Description

The decoding algorithm is described here for Hermitian codes. The Hermitian curve is defined as ,

$$x^m + y^{m-1} + y = 0 \tag{5.2}$$

with $m = q + 1$ in field $\mathbb{F}_{q^2}$. The curve is *maximal* and meets the Hasse-Weil upper bound on the number of points. The algorithm is divided into two parts; determining the values of unknown syndromes and updating minimal polynomials. The syndrome is computed as the two dimensional Fourier transform (FT) of the received sequence, however the symbols are first placed on the points that lie on the curve in the plane prior to the FT. This definition for the syndrome holds true for Blahut's presentation of the Hermitian code and the dual code defined by Justesen (Justesen et al., 1989). For a received vector r of length n, the FT is defined as,

$$S_{(a,b)} = \sum_{i=0}^{n} r_i x_i^a y_i^b \tag{5.3}$$

Since the defining set for the codes is given as $a + b \leq J$, the initial syndromes are given by

$$S_{(a,b)} \quad a + b \leq J$$

all other syndromes are unknown. To correct up to $t = \lfloor \frac{d-1}{2} \rfloor$ errors, the unknown syndromes need to be determined by either using the equation of the defining curve or by majority voting. In a plane, the syndrome components agree with the roots of the curve. For the Hermitian curve in (5.2) the syndromes obey the relationship,

$$\begin{aligned} S_{(a+m,b)} + S_{(a,b+m-1)} + S_{(a,b+1)} &= 0 \\ S_{(a,b)} = S_{(a-m,b+m-1)} + S_{(a-m,b+1)} &= 0 \end{aligned} \tag{5.4}$$

From (5.4) any syndrome element $S_{(a,b)}$ can be determined if $a < m$. For cases where $a > m$ and the syndrome is unknown majority voting is applied. In a simplified form majority voting is just using the set of minimal polynomials $\mathbf{F}$ at some stage of the BMSA to predict (since the polynomials are recursive) the next syndrome and the value with the highest occurrence is chosen as the next syndrome element. For each polynomial $F^i \in \mathbf{F}$

$$\begin{aligned} \sum_{\bar{k}}^{\bar{s}_i} F^i_{\bar{k}} S_{\bar{k}+(\bar{r}-\bar{s}_i)} &= 0 \\ F^i_{\bar{s}_i} S_{\bar{r}} &= \sum_{\bar{k}}^{\bar{s}_i-\bar{1}} F^i_{\bar{k}} S_{\bar{k}+(\bar{r}-\bar{s}_i)} \end{aligned} \tag{5.5}$$

where $\bar{r} = (a,b)$. The polynomials in $\mathbf{F}$ are monic so that $F^i_{\bar{s}_i} = 1$ and $\bar{s}_i - \bar{1}$ represents the bidegree preceding s_i in the chosen monomial order.

$$v_i = S_{\bar{r}} = \sum_{\bar{k}}^{\bar{s}_i-\bar{1}} F^i_{\bar{k}} S_{\bar{k}+(\bar{r}-\bar{s}_i)} \tag{5.6}$$

Equation (5.1) applies to polynomials in $\mathbf{F}$ with span span $= \bar{r} - \bar{s}_i \geq 0$. This simplified majority voting is not sufficient since a vote cannot be decided if all the elements v_i are distinct. A more comprehensive voting scheme was suggested in (Sakata et al., 1995) and utilizes the properties of the BMSA that the size of the footprint Δ at the end of the algorithm does not exceed $t = \lfloor \frac{d-1}{2} \rfloor$ and certain restrictions on the number of polynomials in $\mathbf{F}$ at some stage of the algorithm that do have a nonzero discrepancy. Equation (5.4) can be presented as,

$$S_{(a+m,b-m+1)} = S_{(a,b)} + S_{(a,b-m+2)} \quad b \geq m-1 \tag{5.7}$$

Let $s_i = (s_1, s_2)$ if $a - s_1 + m \geq 0$ and $b - m + 1 - s_2 \geq 0$ then combining (5.7) and (5.1) results in,

$$\sum_{\bar{k}}^{\bar{s}_i-\bar{1}} F_{\bar{k}} S^i_{(k_1+a-s_1+m,k_2+b-m+1-s_2)} + S_{(a,b-m+2)} = w_i \tag{5.8}$$

where $\bar{k} = (k_1, k_2)$. To perform majority voting of unknown syndromes at a stage in the algorithm the sets,

$$\begin{aligned} K_1 &= \{(x,y) | 0 \leq x \leq a \wedge 0 \leq y \leq b\} \\ K_2 &= \{(x,y) | 0 \leq x < m \wedge 0 \leq y \leq b - m + 1\} \\ K &= K_1 \cup K_2 \end{aligned}$$

are formed. Associated with each polynomial $F^i \in \mathbf{F}$ the sets,

$$A_i = \{(x,y) \in K | x + s_1 \leq a \wedge y + s_2 \leq b\}$$
$$B_i = \{(x,y) \in K | x + s_1 \leq a + m \wedge y + s_2 \leq b - m + 1\}$$

Finally values from (5.6) and (5.8), *distinct* values from w_i and v_i obtained from every valid polynomial in $\mathbf{F}$ are selected and form the set of finite field elements $a_1, a_2, \ldots, a_z$ which are candidates for the value $S_{(a,b)}$. For each a_j,

$$P_j = \left(\bigcup_{v_i = a_j} A_i \cup \bigcup_{w_i = a_j} B_i \right) \Big\backslash \Delta_{(a,b)} \tag{5.9}$$

and $S_{(a,b)} = \{a_j : |P_j| = |P_j|_{max}\}$.
For polynomials in the set $\mathbf{F}$ at some stage in the algorithm that have a nonzero discrepancy an update is necessary. A polynomial is not updated if its discrepancy is zero or if it does not *reach* that stage i.e. at some stage $\bar{r} = (a,b)$ the polynomial of degree $\bar{s}_i = (s_1, s_2)$ reaches $\bar{r}$ if $a - s_1 \geq 0$ and $b - s_2 \geq 0$. The algorithm initializes with an empty footprint $\Delta = \varnothing$ and the update rules are given as in Algorithm 5.1.

The BMSA is now illustrated with a specific example. The following set of affine rational points are obtained from the Hermitian curve over $\mathbb{F}_4$

$$\begin{array}{cccc} (0,0) & (0,1) & (\alpha,\alpha) & (\alpha,\alpha^2) \\ (\alpha^2,\alpha) & (\alpha^2,\alpha^2) & (1,\alpha) & (1,\alpha^2) \end{array}$$

The dual Hermitian code defined in Section 4.6.1 has parity check matrix H and and the generator matrix G with $j = 1$ and $d_{min} = 3$ and is a single error correcting code. The G and H matrices are,

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & \alpha & \alpha & \alpha^2 & \alpha^2 \\ 0 & 1 & \alpha & \alpha^2 & \alpha & \alpha^2 & \alpha & \alpha^2 \end{bmatrix}$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \alpha^2 & \alpha^2 & 1 \\ 0 & 1 & 0 & 0 & 0 & \alpha^2 & \alpha & 0 \\ 0 & 0 & 1 & 0 & 0 & \alpha & 1 & \alpha \\ 0 & 0 & 0 & 1 & 0 & \alpha & 0 & \alpha^2 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

G is in reduced form and the reduction process may involve column interchanges in both the H and G matrix. These interchanges need to be reversed in order for the syndrome evaluation to be accurate since the code is not cyclic. The codeword,

$$c = \begin{bmatrix} 0 & 1 & 0 & \alpha^2 & \alpha & 0 & 0 & 0 \end{bmatrix}$$

is chosen and the corrupted vector is

$$r = \begin{bmatrix} 0 & 1 & 0 & \alpha^2 & \alpha & \alpha & 0 & 0 \end{bmatrix}$$

The correspondence between the points of the curve and the coordinates of the codewords according to the points evaluation is given by (5.2). The syndrome can then be computed and is given by Equation (5.3) for the initial syndromes that satisfy $a + b \leq J$. The syndrome is,

Points	Coordinates
$(0,0)$	0
$(0,1)$	1
$(1,\alpha)$	2
$(1,\alpha^2)$	3
(α,α)	4
(α,α^2)	5
(α^2,α^1)	6
(α^2,α^2)	7

Table 5.2: Correspondence between points and coordinates of the Hermitian code

$$S = \begin{bmatrix} \alpha & 1 & * \\ \alpha^2 & * & * \\ * & * & * \end{bmatrix}$$

where * represents the unknown syndromes. The graded lexicographic monomial order in the bivariate ring $\mathbb{F}_4[x,y]$ is given by (5.3).

The algorithm initialises with,

$$\mathbf{F} = \{1\} \quad \mathbf{G} = \varnothing \quad \Delta = \varnothing \quad \text{span}(\mathbf{G}) = \varnothing$$

Index	0	1	2	3	4	5
Bidegree	(0,0)	(1,0)	(0,1)	(2,0)	(1,1)	(0,2)
Monomial	1	x	y	x^2	xy	y^2

Table 5.3: Graded lexicographic order in $\mathbb{F}_4[x,y]$

and at stage 0, $\bar{r}=(0,0)$ and degree of F^1 is $\bar{s}_1=(0,0)$ so that

$$\bar{r}-\bar{s}_1=(0,0)-(0,0)=(0,0) \quad (F^1 \text{ reaches } \bar{r})$$

the discrepancy is calculated as

$$\begin{aligned}
\delta_1 &= \sum_{\bar{k}}^{\bar{s}_1} F^1_{\bar{k}} S_{\bar{k}+(\bar{r}-\bar{s}_1)} \\
\delta_1 &= \sum_{\bar{k}}^{(0,0)} F^1_{\bar{k}} S_{\bar{k}+(0,0)} \\
\delta_1 &= F^1_{(0,0)} S_{(0,0)} \\
\delta_1 &= 1\cdot\alpha=\alpha
\end{aligned}$$

the polynomial F^1 has a nonzero discrepancy and needs to be updated according to the rules of Algorithm 5.1. Since $\bar{r}-\bar{s}_1 \notin \Delta_{(0,0)}$ the rule of line 9 of Algorithm 5.1 is applied. $\bar{r}-\bar{s}_1=(0,0)$ is appended to the footprint so that $\Delta_{(1,0)}=\{(0,0)\}$ which changes the bidegrees and number of polynomials in $\mathbf{F}$.

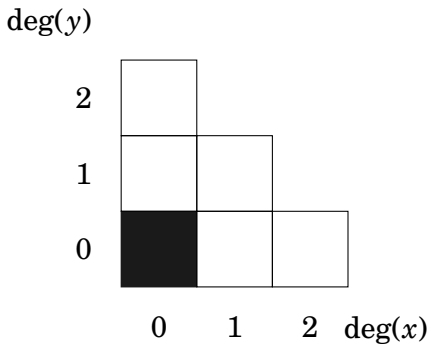

Fig. 5.1: Before update at stage 0

The blackened rectangles in Figures 5.1 and 5.2 are the bidegrees of the leading monomials in $\mathbf{F}$ while the grey rectangles represent the bidegrees of polynomials in the set $\mathbf{F}$. Appending $(0,0)$ to the footprint expands it so that the new bidegrees of the minimal polynomials are $(1,0)$ and $(0,1)$ therefore the update needs to take into consideration this fact. For the first new bidegree $\acute{\bar{s}}_1=(1,0)$, $(q_1,q_2)=\acute{\bar{s}}_1-\bar{s}_1=$

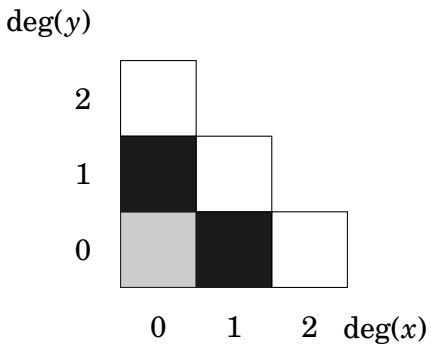

Fig. 5.2: After update at stage 0

$(1,0)-(0,0)=(1,0)$ as in line 13 of 5.1,

$$\acute{F}^i = x^{q_1} y^{q_2} F_i + \frac{\delta_i}{\delta_j} x^{p_1} y^{p_2} G_j$$

and since $\mathbf{G} = \varnothing$, $G_j = 0$. The update is,

$$\acute{F}^1 = x^1 y^0 \cdot 1 = x.$$

Similarly, for the bidegree $\acute{s}_2 = (0,1)$, $(q_1, q_2) = \acute{s}_2 - \bar{s}_1 = (0,1) - (0,0) = (0,1)$ and an update polynomial,

$$\acute{F}^1 = x^0 y^1 \cdot 1 = y.$$

The set $\mathbf{G}$ also needs to be updated. Since the polynomial $F^1 = 1$ has the required bidegree and has a nonzero discrepancy it is appended to $\mathbf{G}$. The span of F^1 , $\text{span}(F^1) = \bar{r} - \bar{s}_1 = (0,0)$ is then stored and also its discrepancy δ_1 by premultiplying,

$$G_1 = \frac{1}{\delta_1} F^1$$
$$G_1 = \frac{1}{\alpha} = \alpha^2$$

At stage 1, $\bar{r} = (1,0)$ and the algorithm sets are

$$\mathbf{F} = \{x, y\} \quad \mathbf{G} = \{\alpha^2\} \quad \text{span}(\mathbf{G}) = \{(0,0)\} \quad \Delta_{(1,0)} = \{(0,0)\}$$

The polynomials in $\mathbf{F}$ are tested to see if they reach $\bar{r}$,

$$\bar{r} - \bar{s}_1 = (1,0) - (1,0) = (0,0)$$
$$\bar{r} - \bar{s}_2 = (1,0) - (0,1) = (1,-1)$$

Only $F_1 = x$ reaches $\bar{r}$. The discrepancy of this polynomial is,

$$\delta_1 = 0 \cdot S_{(0,0)} + 1 \cdot S_{(1,0)} = \alpha^2$$

and $\bar{r} - \bar{s} = (0,0)$ is already in the footprint therefore there is no need to expand it. The update for polynomial F_1 then follows line 7 of Algorithm 5.1,

$$\acute{F}_1 = x + \alpha^2 \cdot G_1 = x + \alpha^2 \cdot \alpha^2 = x + \alpha$$

Aside from the fact that $G_1 = \alpha^2$ is the only polynomial in $\mathbf{G}$, it satisfies the criterion $\bar{s}_1 - (\bar{r} - span(G_1)) \geq (0,0)$. The new sets are,

$$\mathbf{F} = \{x + \alpha, y\} \quad \mathbf{G} = \{\alpha^2\} \quad \text{span}(\mathbf{G}) = \{(0,0)\} \quad \Delta_{(0,1)} = \{(0,0)\}$$

At stage 2, $\bar{r} = (0,1)$ and only $F_2 = y$ reaches $\bar{r}$ with discrepancy,

$$\delta_2 = 0 \cdot S_{(0,0)} + 0 \cdot S_{(1,0)} + 1 \cdot S_{(0,1)} = 1 \cdot 1 = 1$$

with $\bar{r} - \bar{s}_2 = (0,0)$ which is already in the footprint. F_2 is updated by,

$$\acute{F}_2 = y + 1 \cdot G_1 = y + \alpha^2$$

The new sets are,

$$\mathbf{F} = \{x + \alpha, y + \alpha^2\} \quad \mathbf{G} = \{\alpha^2\} \quad \text{span}(\mathbf{G}) = \{(0,0)\} \quad \Delta_{(2,0)} = \{(0,0)\}$$

At stage 3, $\bar{r} = (2,0)$ the syndrome element $S_{(a,b)}$ is unknown, the majority voting scheme is used and $S_{(a,b)} = 1$ emerges from the votes. Only polynomial $F_1 = x + \alpha$ reaches $\bar{r}$ with $\bar{r} - \bar{s}_1 = (1,0)$ which is not in the footprint. The discrepancy of F_1 is $\delta_1 = 0$ therefore no update is necessary and the sets are unchanged.
At stage 4, $\bar{r} = (1,1)$ and the syndrome $S_{(1,1)}$ is unknown. Recall that for an unknown syndrome element $S_{(a,b)}$, if $a < m$ the majority voting procedure is used to determine$S_{(a,b)}$. Only one candidate emerges from the votes and $S_{(a,b)} = \alpha$. Using the new syndrome value decoding proceeds as before and checks if polynomials in $\mathbf{F}$ reach $\bar{r}$. Both polynomials in $\mathbf{F}$ reach $\bar{r}$ and both have zero discrepancies , even though $\bar{r} - \bar{s}_1 = (0,1) \notin \Delta_{1,1}$, the footprint is not expanded and the sets remain unchanged.
For the remaining stages of the algorithm all the syndromes can be computed using majority voting or the equation of the curve and the polynomials in $\mathbf{F}$ remain unchanged since they have zero discrepancies at every stage. The common root of the two polynomials in $\mathbf{F}$ at the end of the algorithm is the root (α, α^2) which from Figure 5.2 corresponds to co-ordinate 5 of the received vector indicating an error has occurred there.

Algorithm 5.1 Update for BMSA

Require: At some stage $\bar{r} = (a, b)$ of the BMSA

for $i = 1 : |\mathbf{F}|$ **do**
  if $\bar{r} - \bar{s}_i \geq (0,0)$ **then**
    Calculate the discrepancy δ_i of F_i
    if $\delta_i \neq 0$ **then**
      if $\bar{r} - \bar{s}_i \in \Delta_{(a,b)}$ **then**
        Set $(p_1, p_2) = \bar{s}_i - (\bar{r} - span(G_j)) \geq (0,0)$ for some $G_j \in \mathbf{G}$
        Update with $\acute{F}^i = F^i + \frac{\delta_i}{\delta_j} x^{p_1} y^{p_2} G_j$
      else if $\bar{r} - \bar{s}_i \notin \Delta_{(a,b)}$ **then**
        Append $\bar{r} - \bar{s}_i$ into $\Delta_{(a,b)}$
        Calculate new bidegrees bideg{$\acute{\mathbf{F}}$}
        for Every $\acute{s}_k \in$ bideg($\acute{\mathbf{F}}$) **do**
          if $\acute{s}_k - \bar{s}_i \geq (0,0)$ **then**
            Set $\acute{s}_k - \bar{s}_i = (q_1, q_2)$ and $(p_1, p_2) = \acute{s}_i - (\bar{r} - span(G_j)) \geq (0,0)$ for some $G_j \in \mathbf{G}$
            Update $\acute{F}^i = x^{q_1} y^{q_2} F_i + \frac{\delta_i}{\delta_j} x^{p_1} y^{p_2} G_j$
          end if
        end for
      end if
    end if
  end if
end for

5.2.1.3 Finding Error Values For BMSA

The error polynomial is given by

$$e(x,y) = \sum_{0}^{t-1} e_u x^r y^s$$

where $\{r,s\}$ are the powers of the error locations (common roots) obtained from the BMSA and $e(x,y)$ has exactly $t = \lfloor \frac{d-1}{2} \rfloor$ nonzero coefficients. The error values e_u can be found from using any t independent equations of the $n-k$ possible equations,

$$e(\alpha^a, \alpha^b) = \sum e_u \alpha^{ra} \alpha^{sb} = \sum_{i=0}^{n} r_i x_i^a y_i^b = S_{(a,b)} \qquad a+b \leq J \tag{5.10}$$

Solving the t equations for e_u then gives the error values. Using equation (5.10) is straightforward if the error locations are in the bicyclic plane. Recall the bicyclic plane is a subset of an affine plane with points having no zero coordinates.
For the previous example $t = 1$ and the error location (α, α^2) is in the bicyclic plane. We have the bivariate error polynomial as,

$$e(x,y) = e_0 x y^2$$

since the error location from the BMSA is (α, α^2). Choosing $a = b = 0$ we have,

$$\begin{aligned} e(\alpha^0, \alpha^0) = e_0 &= S_{(0,0)} \\ &= \alpha \end{aligned}$$

thus the error vector is

$$e = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \alpha & 0 & 0 \end{bmatrix}.$$

Only bicyclic Hermitian codes are treated in Section 5.5.2 and for these codes the procedure above is sufficient to obtain the error values. For the case of Hermitian codes with error locations that have a zero coordinate (i.e. defined in the affine plane), Liu (1999) presents a procedure for finding these error values. The method requires knowledge of syndromes,

$$S_{(a,b)} \qquad a \leq q-1, b \leq q-1$$

Also for affine AG codes Leonard (1996) presents a generalized Forney formula for finding the error values.

5.2.1.4 Decoding Reed Solomon Codes

The algorithm (Sakata, 2010) is given in detail by Algorithm 5.2. In Section 4.6.1 it is shown how Reed Solomon codes can be constructed as AG codes. In this section the classical decoding of Berlekamp and Massey as applied to RS codes is presented. Terminology already established in Section 4.6.1 where decoding of AG codes was implemented in two dimensions is used. This style of presentation follows (Sakata, 2010). The Berlekamp Massey Algorithm (BMA) can be viewed as a one dimensional version of the SBMA. A key difference between the two algorithms is that in the BMA the syndromes of a received sequence are sufficient to completely determine the error locator polynomial whereas in the SBMA more syndromes need to be found using majority voting in order to get an error polynomial with the required roots. The single minimal polynomial f and an interior polynomial g are defined first. Other important parameters are given as,

$$s = \deg(f) \ \ ,\delta_f = \text{discrepancy}(f) \ \ ,\delta_g = \text{discrepancy}(g), \ \ \text{span}(g), \ \ \Delta = \text{footprint}$$

The syndrome vector is given as the FT of the received sequence,

$$S = \sum_{i=0}^{n-1} r_i \omega^{(ij)} \quad j = 0, 1, .., n-1$$

where ω is an nth root of unity in the field $\mathbb{F}_q$. The discrepancy of a minimal polynomial at stage r is calculated using the recursive relationship,

$$\delta_f = \sum_{i=0}^{s} f_i S_{i+(r-s)}$$

The BMA decoding on the $(7,5,3)_8$ single error correcting RS codes is used as an example. Let $\mathbb{F}_8$ be a finite field defined with primitive polynomial $x^3 = x+1$ with the element α as a root. The roots of a generator polynomial of the code as $\{1, \alpha\}$ are chosen. Suppose the transmitted codeword $\bar{c}$, the received vector $\bar{r}$ and the error vector $\bar{e}$ are given as

$$\bar{c} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \alpha & \alpha^3 \end{bmatrix}$$
$$\bar{r} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & \alpha & \alpha^3 \end{bmatrix}$$
$$\bar{e} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The syndrome vector is given by,

$$S = \begin{bmatrix} 1 & \alpha & * & * & * & * & * \end{bmatrix}$$

where the syndrome values denoted by * are not relevant since they are not indexed by the orders of the roots of the code. At step $r = 0$,

$$\delta_f = \sum_{i}^{s} f_i S_{i+(r-s)}$$
$$\delta_f = 1 \cdot 1 = 1$$

and $\text{span}(f) = r - s = 0 - 0$ is not in the footprint Δ, therefore the update rule of Algorithm 5.2 line 9 is applied. Since 0 is the maximum element in the footprint $\acute{s} = 0 + 1 = 1$

$$\acute{f} = x^{1-0} = x$$

At step $r = 1$ the discrepancy is,

$$\delta_f = 0 \cdot 1 + 1 \cdot \alpha = \alpha$$

and $\text{span}(f) = r - s = 1 - 1 = 0$ is already in the footprint. The update for this case is given by line 5 of the algorithm,

$$\acute{f} = x + \frac{\alpha}{1} x^{1-(1-0)} \cdot 1$$
$$= x + \alpha$$

The algorithm terminates here since the number of roots of f is equal to the error correction limit of the code $t = 1$. The root of f is α which has order 1 and therefore points to index 1 of the received vector as an error. The error value can be obtain by simply evaluating the received polynomial at the inverse of the error location α.

$$r(\alpha^{-1}) = 1 + \alpha^{-1} + \alpha^{-5} \cdot \alpha + \alpha^{-6} \cdot \alpha^3$$
$$= 1$$

5.3 Maximum Likehood Erasure Decoding

A linear code of minimum distance d is guaranteed to correct $d - 1$ erasures but will also correct some erasure patterns greater than $d - 1$ (MacWilliams and Sloane, 1983). An analysis of the number of correctable error patterns was presented by (Tomlinson et al., 2007) and was found to be dependent on the weight distribution of low weight codewords. The procedure is akin to the erasure method of finding the minimum distance of linear code described in Section 2.5.1. The first steps involve testing for solvability in column co-ordinates of the parity check matrix that correspond to the erased symbols. If the erasures are solvable the erasures are then solved using back substitution or any other means of solving homogeneous

equations. On the other hand, unsolvable erasure patterns are left unaltered. The procedure is described algorithmically in 5.3. At the step 9 of the algorithm, the submatrix formed by the columns of **H** corresponding to the positions of erasures and the rows of **H** will be in upper triangular form.

5.4 Ordered Reliability Decoding

In erasure decoding an error correction capability of $d_{min}-1$ erasures is guaranteed however sometimes more errors can be corrected. However in error correction of codes, bounded distance decoding only guarantees $\left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ errors that can be corrected. In errors-only decoding a natural disadvantage is that the decoder has no knowledge of the error locations. In an ideal scenario it would be desirable to have the locations of error and each error could be treated as an erasure and correct many more errors. However this is not possible. A midway solution is to use channel information and attempt to determine the reliability of the individual elements of the received sequence and based on some preferred criterion select the least reliable elements and treat as erasures. Estimates of reliability from channel information are not perfect and in soft decision decoding of linear block codes the decoding procedure for each received sequence will have to be repeated a number of times so that a list of candidate codewords is created with each decoding. A codeword from the list that minimizes error is then chosen.
In AWGN channels using binary phase shift keying the reliability of the elements of the received sequence is known to be proportional to the a-priori likelihood ratio. At the ith position of the received sequence $\bar{y}$ given that a codeword $\bar{c}$ was sent the log-Likehood ratio is given,

$$\begin{aligned} L_i &= log\left(\frac{P_r(y_i|\acute{c}_i=0)}{P_r(y_i|\acute{c}_i=1)}\right) \\ &= log\left(\frac{\frac{1}{\sqrt{2\pi}\sigma}e^{-(y_i+1)^2/2\sigma^2}}{\frac{1}{\sqrt{2\pi}\sigma}e^{-(y_i-1)^2/2\sigma^2}}\right) \\ &= -\frac{2}{\sigma^2}y_i \end{aligned} \tag{5.11}$$

therefore the probability of the ith element of the received sequence $\bar{y}$ being correct is proportional to $|y_i|$ (Tjhai, 2007). Reliability based soft decision procedures utilize these reliability values to order the received sequence $\bar{y}$ in order of reliability and attempt to correct errors. Notable reliability based decoding algorithms are the generalized minimum distance decoding introduced by Forney, the Chase type algorithms and the ordered statistics decoding. The first two algorithms exploit the fact a linear code can correct up to ($2\times$number of errors + number of erasures $\leq d_{min}-1$) combinations of errors and erasures while the latter exploits an earlier observation that a code can correct up to $d_{min}-1$ erasures and sometimes more. Soft deci-

sion decoding using ordered reliability decoding is essentially the ordered statistics decoder of Fossorier and Lin (1995) without the statistics and optional stopping criterion suggested therein.

The procedure is described assuming binary codes and then later adjustments are made to non-binary codes. A code with message length k and length n has a received/corrupted sequence $\bar{y}$ is ordered according to decreasing reliability $|y_i|$ so that a new sequence $\bar{z}$ has $r_i > r_{i+1}$ for all i. The ith column of the generator matrix G is also ordered according to the reliability of y_i . The new generator matrix $G_{\bar{z}}$ is then expressed in reduced row echelon form by Gaussian elimination and if column interchanges are necessary then these index changes are applied on the sequence $\bar{z}$ to produce a new sequence $\bar{b}$ and a new generator matrix $G_{\bar{b}}$. The real valued sequence $\bar{b}$ is then hard decided using BPSK demodulation into binary values and and a bit valued sequence $\bar{s}$ is formed. The sequence is then partitioned into two; the most reliable part (MRP) which includes the first k most reliable symbols and the least reliable part (LRP) which is the $n-k$ least reliable symbols of the sequence. Decoding then involves deleting the $n-k$ least reliable positions of $\bar{s}$ and re-encoding using $G_{\bar{b}}$. If the number of errors in the sequence is $\leq d-1$ and the reliability measure is accurate enough to guess all the error locations, then the new codeword is the codeword closest to the received codeword. However these two conditions are not always met and additional reprocessing is needed in order to eliminate errors in the MRP. This reprocessing is simply subtracting error vectors systematically from the received binary sequence $\bar{s}$ and re-encoding until codeword with minimum euclidean distance from $\bar{s}$ is found. For all combination of i errors in the MRP results in,

$$\binom{k}{i}$$

possible error vectors.

There are two ways to implement ordered reliability decoding on non-binary codes; firstly by using the binary image expansion of the code and secondly by using symbol based decoding. Symbol based decoding is used in this case and some slight adjustments are made to the procedure. Firstly, the symbols of the codeword in the field $\mathbb{F}_{q^m}$ prior to transmission are mapped to binary using a suitable basis so that each unique element is represented by m bits. Re-encoding is done in $\mathbb{F}_{q^m}$. Since each symbol is represented by m bits, it is also represented by m reliability values. In order to sort the symbols of the received sequence according to reliability, for each symbol a *representative* reliability value is chosen from the m possible choices. A natural choice of a representative is the element from the m values with the least reliability since it only takes a single value to be in error in an m block in order for the symbol the block represents to also be in error. These chosen reliability values now represent the m blocks and consequently a symbol in the sequence and can be

used to order the sequence of symbols as in the binary case. Euclidean distance is however still measured using the binary representation of the codewords.
For the case of non-binary codes the reprocessing involves non-binary symbols as well and for all combinations of i errors results in,

$$\binom{k}{i}(q-1)^i$$

possible error vectors. Reprocessing for all possible combinations so as to achieve maximum likelihood decoding is difficult and will involve,

$$\sum_{i=0}^{k}\binom{k}{i}(q-1)^i$$

possible error vectors. Therefore the procedure is terminated after J possible candidate codewords have been produced and then choose one with the smallest euclidean distance from the received sequence as the most likely codeword.

Algorithm 5.2 Berlekamp Massey Algorithm

Require: Syndrome, S

Initialize $r = 0$, $f = 1$,$g = 0$,span(g)=-1, $\Delta = \varnothing$
Compute the discrepancy δ_f
if $\delta_f \neq 0$ **then**
  if span(f)=$r - s \in \Delta$ **then**
    Update $\acute{f} = f + \frac{\delta_f}{\delta_g} x^{s-(r-\text{span}(g))}$
  end if
  if span(f)=$r - s \notin \Delta$ **then**
    Append $r - s$ to Δ
    Let l be the largest element in Δ, set $\acute{s} = l + 1$
    Update $\acute{f} = x^{\acute{s}-s} + \frac{\delta_f}{\delta_g} g$
    Set $g = f$, span(g) = span(f)
  end if
end if
r=r+1
if number of roots of f =t **then**
  Stop
else
  Go to 2
end if

Algorithm 5.3 Maximum Likehood Erasure Decoding

Require: $\mathbf{H}$, $c_1, c_2, \ldots, c_e$=erasure positions

for $i : e$ **do**
  Choose co-ordinate $\mathbf{H}_{i,c_i}$ of the $\mathbf{H}$ matrix
  if $H_{i,c_i} = 0$ and $\exists H_{k,c_i} \neq 0 \; \forall \; k > i$ **then**
    Interchange row i with row k of $\mathbf{H}$
  else if $H_{i,c_i} = 0$ and $\nexists H_{k,c_i} \neq 0 \; \forall \; k > i$ **then**
    Exit {erasures cannot be solved}
  end if
  Perform gaussian elimination on the rows of $\mathbf{H}$ so that all positions $\mathbf{H}_{k,c_i} = 0 \; \forall \; k > i$
end for
Use reduced $\mathbf{H}$ to solve for the erasures

Algorithm 5.4 Ordered reliability decoding for non-binary codes

Require: A received sequence $\bar{a}$, generator matrix G

1: For each m block in $\bar{a}$ associate a reliability value r_i corresponding to the smallest reliability value
2: Perform BPSK demodulation on $\bar{a}$ to form a binary sequence $\bar{b}$
3: Form a sequence of symbols $\bar{v}$ with elements from $\mathbb{F}_{q^m}$ from $\bar{b}$ using the selected basis
4: Order $\bar{v}$ in order of decreasing reliability , apply the corresponding changes to columns of G and the m blocks of $\bar{a}$
5: Express G in systematic form and apply the same column interchanges that occurred as result (if any) to $\bar{v}$
6: **for** $l = 0 : i$ **do**
7: **for** $j = \binom{k}{i}(q-1)^i$ **do**
8: Subtract an error vector of weight l from $\bar{v}$ to form $\bar{c}$
9: Delete the $n-k$ least reliable positions of $\bar{c}$ and re-encode with G to form $\bar{w}$
10: Store the euclidean distance between the BPSK modulation of $\bar{w}$ and the ordered $\bar{a}$
11: **end for**
12: **end for**
13: Select the codeword with the minimum euclidean distance

5.5 Performance of Algebraic Geometry Codes

Performance comparison of AG codes and RS codes in previous literature (Johnston and Carrasco, 2005) compare these two types of codes defined in the same finite field and having similar rates but the codes have unequal lengths since RS codes are by definition shorter. Subfield subcodes of RS codes (namely BCH codes) contain all those codewords of an original RS code that have alphabets in a smaller field. In this aspect, AG codes and BCH codes are similar. In comparing AG and BCH codes, the shortening of BCH codes may be necessary so that the codes have equivalent lengths without loss in performance. An epicyclic Hermitian[1] code defined in $\mathbb{F}_{q^2}$ has length $n = q(q^2-1))$ with message length k has minimum distance $d = n-k-g+1$ where g is the genus of the defining curve. Table 5.4 shows the parameters of some Hermitian codes.

A Reed Solomon code of codeword length n and message length k is described using a defining set which is a set of $n-k$ consecutive elements of a finite field as roots of its generator polynomial. BCH codes are formed by first defining a smaller field and then picking those codewords of the RS code that have elements only in that field. For each root of an RS generator polynomial, a BCH code generator polynomial will have additional roots that fall within the same conjugacy class in the smaller field. This has the effect of reducing the rate of the code but not increasing the minimum distance accordingly. The result is that RS codes have greater minimum distances than BCH codes. BCH codes therefore have minimum distance $d = n-k-\delta+1$ where $\delta > 0$. Table 5.5 shows some BCH codes shortened to have equal lengths and rates as the Hermitian codes. In the comparison, the $(60,45,10)_{16}$ bicyclic hermitian code is chosen and its performance compared with the $(60,45,9)_{16}$ BCH code over the additive white Gaussian noise AWGN and the erasure channel.

5.5.1 Encoding

The dual of the hermitian code defined by the divisor $D = 20P_\infty$ and divisors $T = P_1 + \cdots + P_{60}$ in $\mathbb{F}_{16}$ where P_i are the 60 points of the hermitian code in the bicyclic

[1] Epicyclic hermitian codes are hermitian codes defined in the bicyclic plane.

q	n	k	d
4	60	50	5
4	60	45	10
4	60	40	15
8	504	432	45

Table 5.4: Epicyclic Hermitian codes

q	n	k	n_{short}	k_{short}	d
4	255	245	60	50	6
4	255	240	60	45	9
4	255	235	60	40	11
8	4096	4024	504	432	37

Table 5.5: BCH codes

plane is used. The Hermitian code has parameters $[60,15,40]_{16}$ while its dual has parameters $[60,45,10]_{16}$. The parity check matrix of the dual Hermitian code is given by,

$$\mathbf{H} = \begin{bmatrix} f_1(P_1) & f_1(P_2) & \dots & f_1(P_n) \\ f_2(P_1) & f_2(P_2) & \dots & f_2(P_n) \\ . & . & . & . \\ . & . & . & . \\ f_{15}(P_1) & f_{15}(P_2) & \dots & f_{15}(P_n) \end{bmatrix}$$

where $\{f_1,\dots,f_{15}\}$ are basis rational functions of the Riemann-Roch space $L(20P_\infty)$. For the BCH code an RS code is defined in $\mathbb{F}_{2^8}$ and the subfield subcode in $\mathbb{F}_{2^4}$ is found. The subfield subcode is chosen to have a similar rates to the Hermitian code. The Hermitian code has redundancy $n-k=15$ and this number of roots is easily chosen for the generator polynomial of the BCH code. From Section 2.1 it is known that conjugacy classes with at most $\frac{8}{4}=2$ members are obtainable. The first 8 conjugacy classes will suffice,

$$\begin{array}{cccc} (1) & (\alpha,\alpha^{16}) & (\alpha^2,\alpha^{32}) & (\alpha^3,\alpha^{48}) \\ (\alpha^4,\alpha^{64}) & (\alpha^5,\alpha^{80}) & \alpha^6,\alpha^{96}) & (\alpha^7,\alpha^{112}). \end{array}$$

The union of these classes has eight consecutive finite field elements and the BCH code generated by a polynomial with the union as roots will have designed $d_{min} = 8+1=9$. This will produce a $(255,240,9)_{16}$ code. However the symbol elements of the codewords in the new code are still in $\mathbf{F}_{2^8}$ but are isomorphic to $\mathbb{F}_{2^4}$ and a choice of a defining primitive polynomial for the field $\mathbb{F}_{2^4}$ that preserves this isomorphism is $x^4+x+1=0$. The elements of the two fields that are present in the codewords symbols can now be mapped one to one. Finally the code is shortened by deleting 195 information symbols to $(60,45,9)_{16}$.

5.5.2 AWGN Channel with Hard Decision Decoding

The general procedure used for hard decision decoding both codes is explained here in detail. The symbols of each codeword is represented by $m=4$ bits using the basis

defined by the primitive polynomial used to obtain the field $\mathbb{F}_{2^4}$. The binary bits are then modulated using binary phase shift keying (BPSK) in which the mapping from bits to real values is given by $[1] \rightarrow [-1]$ and $[0] \rightarrow [1]$. The modulated sequence is passed through a simulated AWGN channel with variance σ^2 and mean $\mu = 0$. The ratio of energy per bit to the noise spectral density E_b/N_o is specified in decibels dB and the mapping to the channel variance is given by,

$$\sigma^2 = \frac{1}{2 \cdot \frac{E_b}{N_o} \cdot rate}$$

The sequence output from the channel is then demodulated using BPSK demodulation so that if the sequence is of length l and is denoted by v then, a new binary sequence w is

$$w_i = \begin{cases} 1 & \text{if } v_i < 0 \\ 0 & \text{if } v_i \geq 0 \end{cases}$$

The sequence w is then partitioned into $l/4$ parts of 4 bits each and each part is mapped to field symbols using a basis defined by the primitive polynomial used obtain the field. In order to test for a errors, a syndrome test is carried out to check for the presence of errors. If errors are found the received sequence is passed to the appropriate hard decision decoder which attempts to correct the errors. The entire process is repeated a number of times for each E_b/N_o value until 100 symbol errors are encountered and the probability of error is then computed. The results in Figure 5.3 show that the performance of the two codes is similar in the AWGN channel. This is expected because the error correction capability of the two codes is the same,

$$t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$$
$$t_{HER} = \left\lfloor \frac{10 - 1}{2} \right\rfloor = 4$$
$$t_{BCH} = \left\lfloor \frac{9 - 1}{2} \right\rfloor = 4$$

The theoretical probability of frame error is obtained from the following expression (Peterson and Weldon, 1972),

$$P_f = 1 - (1 - P_e)^{(45 \times 4)}$$

where P_e is the probability of bit error for BPSK modulation in the AWGN channel given by (Proakis, 2008),

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_o}}\right)$$

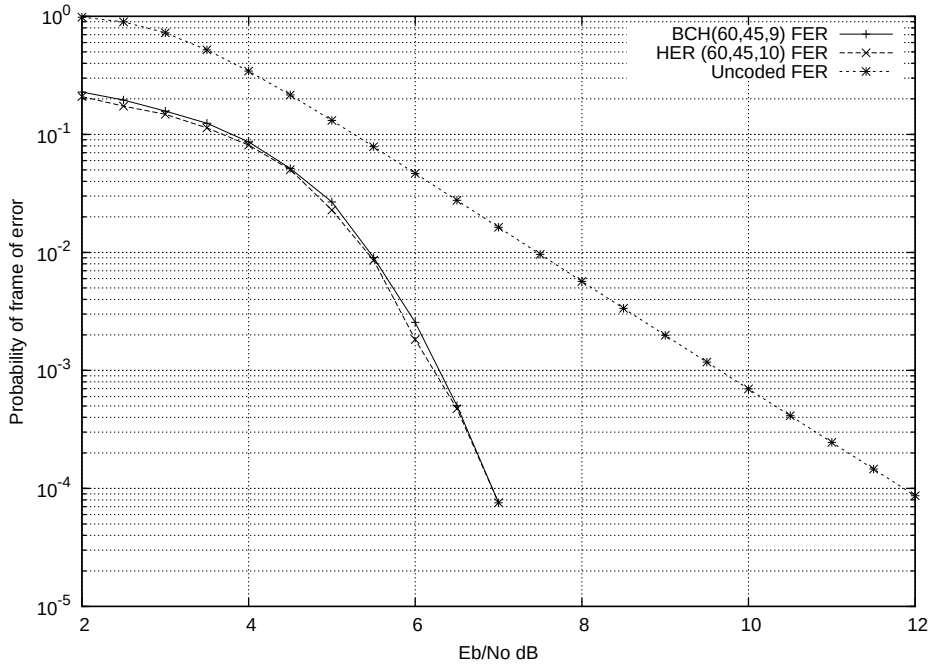

Fig. 5.3: Hard Decision Decoding for Hermitian and BCH Codes

It is also worth mentioning that the BCH codes are decoded in the field $\mathbf{F}_{256}$ since they do not contain any meaningful roots in the field $\mathbf{F}_{16}$ (and will not satisfy the syndrome equations).

5.5.3 Erasure Channel

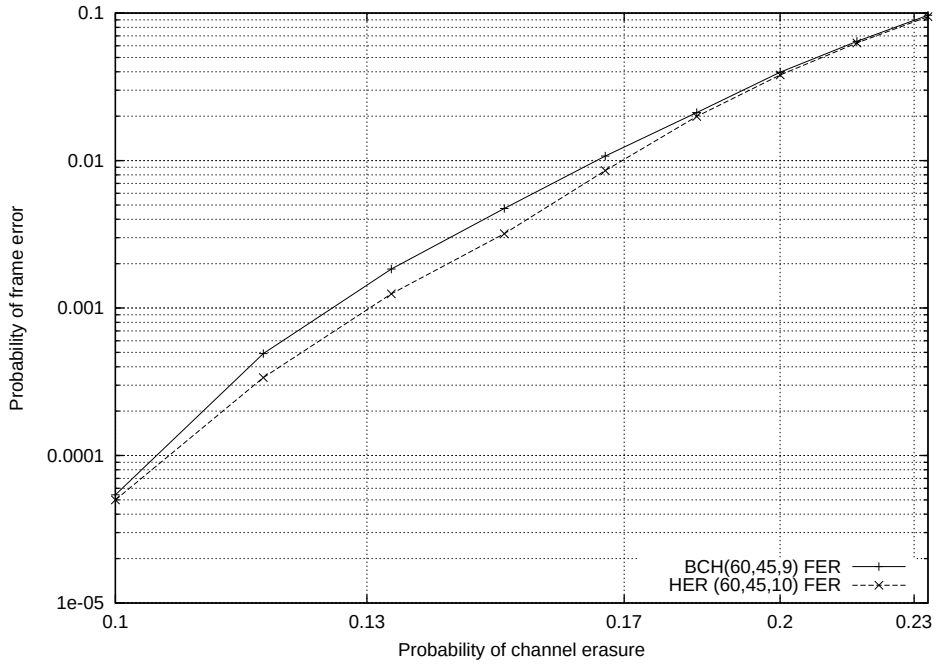

Fig. 5.4: Erasure Decoding for Hermitian and BCH Codes

In simulating the erasure channel the channel erasure probability p and a real valued uniform random generator producing values within the range $0 \leq x \leq 1$ sim-

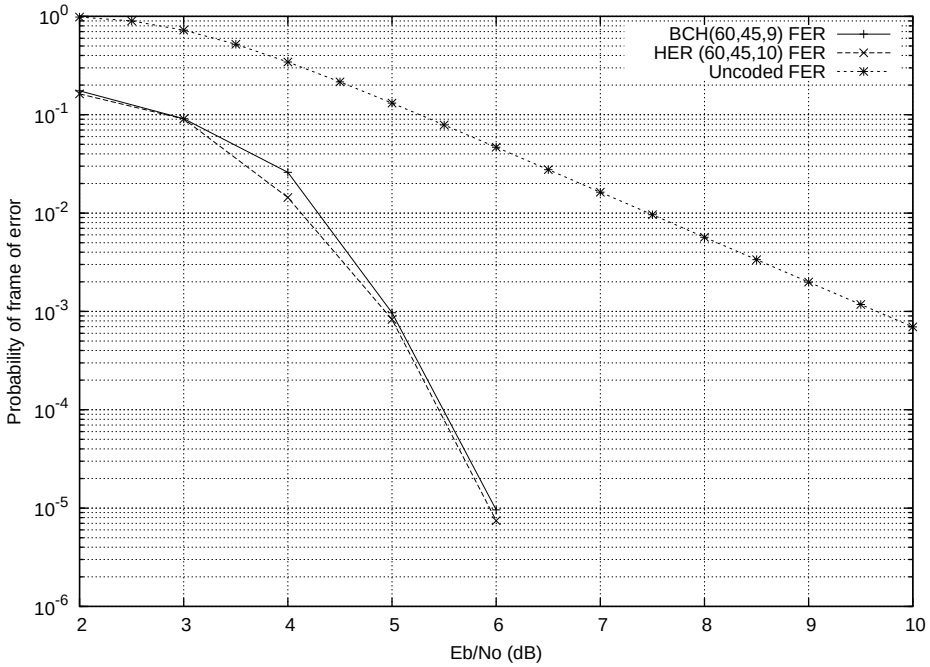

Fig. 5.5: Ordered Reliability Decoding for Hermitian and BCH Codes at order 2

ulates the randomness in the channel are specified. The codeword symbols are erased when the randomly generated value x falls within the range $1-p \leq x \leq 1$ and left are unaltered otherwise. The corrupted codewords are then decoded with Algorithm 5.3 which tests for solvability . If the erasures can be corrected the solution to the homogeneous equations is determined otherwise the corrupted codeword is left unaltered. For each chosen channel probability p the procedure is repeated until 100 erasures are unsolved and the frame erasure/error rate is computed. The performance of hermitian and BCH codes is compared in Figure 5.4 . performance of the codes that the hermitian codes have a performance that surpasses the BCH codes. In the region where the probability of channel erasure is low i.e. just around $p = 0.1$ it is clear that the performances of the two codes are similar since the number of erasures are likely to be below the erasure correction capability of the codes. However at a very large probability of erasure (above $p = 0.2$) the average number of erasures exceeds the error correction capability of both codes and thus their performances are similar. In the region in between the Hermitian code has better performance than the BCH code.

5.5.4 AWGN channel with Soft Decision Decoding

The ordered reliability decoding (symbol based) described in Section 5.4 is used. Figure 5.5 shows the performance of the two codes using ordered reliability decoding with order 2 reprocessing. From the figure the performance of the Hermitian code surpasses the BCH code in particular for energy per bit to noise spectral density ratio within the range $3-5$ dB the Hermitian $(60,45,10)_{16}$ code corrects more

errors than the BCH $(60, 45, 9)_{16}$ code. There is also a slight improvement in performance beyond 5 dB.

5.6 Summary

In this chapter the BMSA for AG codes is presented. The BMSA is a hard decision decoding algorithm that can be used to correct errors up to the bounded distance error correcting limit of the code. The algorithm finds the error locations of a corrupted received vector. In addition the well known Berlekamp Massey decoding algorithm for BCH and RS codes was presented. While the BMSA is an algorithm designed for AG codes, a generic algorithm is presented for erasure correction. This is called the in-place algorithm (Cai et al., 2005) and solves for erasures in an efficient manner. Similarly for soft decision decoding a generic ordered reliability decoding was presented. The performance of non-binary BCH codes and Hermitian codes in the AWGN and erasure channels was compared. The results show that the Hermitian code gives a slightly better performance in the AWGN channel with hard decision decoding and in the AWGN channel with ordered reliability decoding. In the erasure channel the Hermitian code outperforms the non-binary BCH codes for a range of probability of erasure. This behaviour is most likely due to the Hermitian code having a more favorable weight structure. It is difficult however to confirm this as computing the weight distribution of the two codes is difficult.

Part III

Search For New Codes

6. INTRODUCTION

6.1 Maximising the Minimum Distance

Claude E. Shannon's deduced in a nonconstructive way that there exist good linear codes under certain conditions that can help achieve error-free communications over a chosen channel. It did not take long to realize that the definition of a good code is one with a good minimum distance. Consider the case of the binary symmetric channel (BSC) with channel probability p, the probability of decoding error for the channel is given by

$$P_e = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i}$$

for a code of length n and minimum distance d (MacWilliams and Sloane, 1983). The expression for P_e assumes bounded distance decoding with the code able to correct $t = \frac{\lfloor d-1}{2} \rfloor$ errors. For a fixed channel probability p and code length n increasing the minimum distance will increase the number of errors the code can correct and decrease the probability the decoded codeword will be in error. If maximum likelihood hard decision decoding on the BSC is assumed then the expression for P_e becomes,

$$P_e = 1 - \sum_{i=0}^{t} \binom{n}{i} p^i (1-p)^{n-i} - a_{t+1} p^{t+1} (1-p)^{n-t-1}$$

where a_i is the number of coset leaders with weight i (MacWilliams and Sloane, 1983). Again the effect on an increased minimum distance d on P_e is evident. Consider the additive white Gaussian noise channel (AWGN)[1] the union bound on the probability decoding error using maximum likelihood soft decision decoding is given by,

$$P_e \leq e^{-\frac{d}{4N_0}}$$

in a channel with noise spectral density N_0 using a code with minimum distance d (Proakis, 2008). Again the role of the minimum distance d of the code can be observed from the union bound since P_e exponentially decreases with an increasing

[1] A specific type of the BSC when binary transmission is used.

d.

6.2 Tables of Best Known Codes

A fundamental question in coding theory is "*Given a length n, a dimension k and a finite field of cardinality q, what is the best possible minimum distance d obtainable from any code?*". The tables in (Grassl, 2007) and (Schimd and Shurer, 2004) contain upper bounds on the best possible minimum distance of a linear codes with parameters $[n,k,d]_q$. These upper bounds are derived from several different combinatorial bounds. The tables also contain lower bounds of the minimum distance of best known codes. The lower bounds are constructive i.e. there exist known linear codes with parameters $[n,k,d]_q$ for which d has been verified either computationally or mathematically. The goal in this part of the thesis is to obtain codes with a minimum distance greater than that of codes in these tables that have the same length and dimension. The first catalog of best known codes was presented by Cal-

Finite Field	Range
$\mathbb{F}_2$	$1 \le k \le n \le 256$
$\mathbb{F}_3$	$1 \le k \le n \le 243$
$\mathbb{F}_4$	$1 \le k \le n \le 128$
$\mathbb{F}_5$	$1 \le k \le n \le 100$
$\mathbb{F}_7$	$1 \le k \le 10$ $1 \le n \le 50$
$\mathbb{F}_8$	$1 \le k \le 40$ $1 \le n \le 85$
$\mathbb{F}_9$	$1 \le k \le 20$ $1 \le n \le 121$

Table 6.1: Ranges for codes in (Brouwer, 1998)

abi and Myrvaagnes (1964) containing binary codes of length n and dimension k in the range $1 \le k \le n \le 24$. (Sloane, 1972) later presented an updated version of the tables. Helgert and Stinaff (1973) improved the tables in (Calabi and Myrvaagnes, 1964) and presented bounds on binary codes in the range $1 \le k \le n \le 127$. Verhoeff (1987) updated the tables in Helgert and Stinaff (1973) and Brouwer and Verhoeff (1993) subsequently made further updates. The tables at the time contained bounds on binary codes in the range $1 \le k \le n \le 127$. Brouwer (1998) subsequently presented a comprehensive update to the tables which included codes with finite fields up to size 9 with the ranges for k and n given in Table 6.1. At present Grassl (2007) maintains a significantly updated version of the tables in (Brouwer, 1998). The tables now contain codes with k and n in ranges from Table 6.2. A database of the codes in (Grassl, 2007) is included in computer algebra system Magma (Bosma et al., 1997). Finally, Schimd and Shurer (2004) provide a online database for optimal parameters of (t,m,s)-nets, (t,s)-sequences, orthogonal arrays, linear codes,

Finite Field	Range
$\mathbb{F}_2$	$1 \leq k \leq n \leq 256$
$\mathbb{F}_3$	$1 \leq k \leq n \leq 243$
$\mathbb{F}_4$	$1 \leq k \leq n \leq 256$
$\mathbb{F}_5$	$1 \leq k \leq n \leq 130$
$\mathbb{F}_7$	$1 \leq k \leq n \leq 100$
$\mathbb{F}_8$	$1 \leq k \leq n \leq 130$
$\mathbb{F}_9$	$1 \leq k \leq n \leq 130$

Table 6.2: Ranges for codes in (Grassl, 2007)

and ordered orthogonal arrays. These are relatively new tables give the best known codes up to finite fields of size 256. The tables place a restriction on the length of code such that $n \leq 1 \times 10^6$ and put a restriction on $n-k$ for different fields.

6.3 Methodology and Approach

Improvements to the tables in (Grassl, 2007) can be made using an adhoc approach. For any two parameters from n, k and d one searches the tables and identifies possible room for improvement. This room for improvement usually manisfests as a plateau in a plot of the parameters of the codes (within a range) in the tables. Also worth taking into consideration is the gap between the lower bounds and upper bounds for the parameters of the codes in the tables. A large gap suggests that a code can be improved. For example in the finite field $\mathbb{F}_4$ for a fixed dimension $k = 35$, consider the plot of distance d versus length n for $55 \leq n \leq 100$ in Figure 6.1 of codes in (Grassl, 2007). The plot shows the $[81,35,23]_4$ code which is at one edge of a plateau with the second edge at $[77,35,23]_4$. There is an increase in length of 4 symbols whilst the distance remains unchanged. Furthermore the gap between the upper and lower bounds on the minimum distance of the code with length 81 and dimension 35 is significant. Clearly there is room for improvement for $n = 81$ and $k = 35$ in $\mathbb{F}_4$. One can then use the different methods of constructing good codes from existing ones for codes of length 81 and dimension 35 to produce improved codes. This approach is quite difficult as the tables in (Grassl, 2007) are well maintained and include computer routines that do this automatically. A different approach is to use an efficient method of producing good codes and utilize computer programs to find codes with better minimum distance than codes in the (Grassl, 2007). This approach is less intuitive and more generic than the adhoc approach. Also as mentioned earlier most of the known methods are incorporated in (Grassl, 2007). In the course of searching for new codes it became clear that methods that extend the length of the code while increasing the distance in some linear manner have the most potential to produce codes that improve on the best known codes. This is because the distance of the codes in the tables does not increase proportionally as the length increases for a fixed dimension. It is also helpful if the

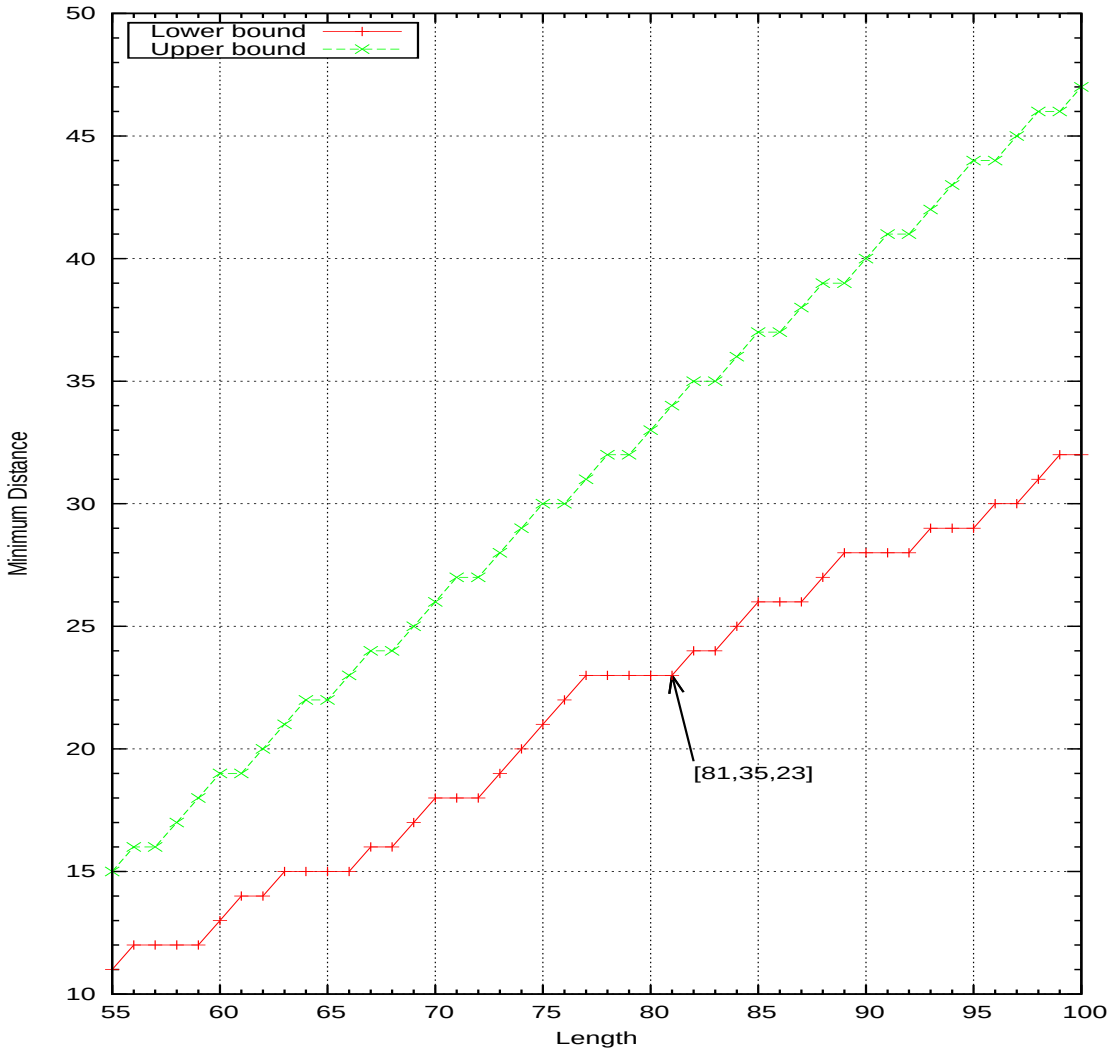

Fig. 6.1: Minimum distance against length for best known codes with dimension 35 *in* $\mathbb{F}_4$

method produces codes with a known lower bound as one need only construct the codes and can skip the tedious procedure of verifying the minimum distance of the code.

At first the search for new codes began by using AG codes. A catalog of good curves was obtained from a table of curves with many curves maintained by Van Der Geer in (Geer et al., 2009). Using these curves, AG codes were constructed and generic code constructions were applied in the search for new codes. Some of the generic constructions include but are not limited to,

- Construction X
- Construction X3
- Construction X4
- Code concatenation
- Blokh-Zyabolov (Zinoviev) generalised code concatenation

- Subfield subcode, subfield image and trace constructions
- $(u, u+v)$ construction
- Grassl special puncturing (Grassl and White, 2004)

Details of these methods can be found in (MacWilliams and Sloane, 1983). No improvements were found using these methods with AG codes. However some best known codes were found. Table 6.3 shows some best known codes from AG codes using the trace construction. Table 6.4 gives some best known codes in $\mathbb{F}_4$ from hermitian codes in $\mathbb{F}_{16}$ using concatenation.

Best Code	Method	AG code	AG Description	Curve	Points	Genus
$[64,41,8]_2$	Trace	$[64,21,30]_8$	AG(34P)	$x^8+x+y^{10}+y^3$	65	14
$[64,44,8]_2$	Trace	$[64,24,24]_8$	AG(37P)	$x^8+x+y^{10}+y^3$	65	14
$[64,53,4]_2$	Trace	$[64,30,21]_8$	AG(43P)	$x^8+x+y^{10}+y^3$	65	14
$[64,56,4]_2$	Trace	$[64,34,17]_8$	AG(47P)	$x^8+x+y^{10}+y^3$	65	14
$[64,57,4]_2$	Trace	$[64,36,15]_8$	AG(49P)	$x^8+x+y^{10}+y^3$	65	14
$[64,13,24]_2$	Trace	$[64,6,54]_{16}$	AG(10P)	x^5+y^4+y	65	6
$[64,55,4]_2$	Trace	$[64,28,31]_{16}$	AG(33P)	x^5+y^4+y	65	6
$[64,59,2]_2$	Trace	$[64,30,29]_{16}$	AG(35P)	x^5+y^4+y	65	6
$[64,63,2]_2$	Trace	$[64,38,21]_{16}$	AG(43P)	x^5+y^4+y	65	6
$[32,13,12]_4$	Trace	$[32,8,23]_{16}$	AG(9P)	$\alpha^{10}x^8+\alpha^5x^4y+\alpha^{10}x^4+x^3y^2+\alpha^5x^2y+\alpha^{10}x^2+\alpha^{10}xy^2+\alpha^5xy+y^4+y^3+y^2$	33	2
$[32,14,12]_4$	Trace	$[32,9,22]_{16}$	AG(10P)	$\alpha^{10}x^8+\alpha^5x^4y+\alpha^{10}x^4+x^3y^2+\alpha^5x^2y+\alpha^{10}x^2+\alpha^{10}xy^2+\alpha^5xy+y^4+y^3+y^2$	33	2
$[32,18,8]_4$	Trace	$[32,11,20]_{16}$	AG(12P)	$\alpha^{10}x^8+\alpha^5x^4y+\alpha^{10}x^4+x^3y^2+\alpha^5x^2y+\alpha^{10}x^2+\alpha^{10}xy^2+\alpha^5xy+y^4+y^3+y^2$	33	2

Table 6.3: Some best known codes from AG codes using Trace Construction

Best code	Inner Code	Outer Code
$[192,38,80]_4$	$[64,19,40]_{16}$	$[3,2,2]_4$
$[195,38,82]_4$	$[65,19,41]_{16}$	$[3,2,2]_4$
$[195,40,80]_4$	$[65,20,40]_{16}$	$[3,2,2]_4$
$[195,42,78]_4$	$[64,21,39]_{16}$	$[3,2,2]_4$

Table 6.4: Best codes in $\mathbb{F}_4$ from Hermitian codes using concatenation

More complicated constructions were then used in larger fields and in particular the generalised AG code construction by Xing et al. (1999a); Xing and Yeo (2007) was used effectively to obtain many new improvements in the finite field $\mathbb{F}_{16}$. Chapter 7 presents these results and give details of the construction. Codes from the database in MAGMA (Bosma et al., 1997) (other than AG codes) were also used together with the aforementioned constructions but only best known codes were found. Figure 6.5 shows some best known codes in $\mathbb{F}_8$ obtained from puncturing best known linear codes (BKLC) in MAGMA. The coordinates of the codes are assumed to be indexed as $[1..n]$. In Chapter 8 many improvements to codes in (Grassl, 2007) in $\mathbf{F}_7$, $\mathbf{F}_8$, and $\mathbf{F}_9$ using a construction of extended Goppa codes are presented.

Best Code	Puncturing Coordinates	BKLC in MAGMA
$[28,10,15]_8$	$[1,2]$	$[30,10,16]_8$
$[31,10,16]_8$	$[1,2,3,4]$	$[35,10,19]_8$
$[86,10,58]_8$	$[1,2,3,4,5,6,8,9,10]$	$[95,10,66]_8$
$[17,10,6]_8$	$[1,2,3,4]$	$[21,10,9]_8$
$[86,10,58]_8$	$[1,2,3,4,5,6,8]$	$[93,10,64]_8$
$[86,10,58]_8$	$[1,2,3,4,5,6,8,10]$	$[94,10,65]_8$
$[86,10,58]_8$	$[1,2,3,4,5,6,8,9,10]$	$[95,10,66]_8$
$[87,10,59]_8$	$[1,2,3,4,5,6,8,9,10]$	$[96,10,67]_8$
$[87,10,59]_8$	$[1,2,3,4,5,6,7,8,9,10]$	$[97,10,68]_8$
$[92,10,63]_8$	$[1,2,3,4,5,6,8,12]$	$[100,10,70]_8$
$[92,10,63]_8$	$[1,2,3,4,5,6,12]$	$[99,10,69]_8$
$[92,10,63]_8$	$[1,2,3,4,5,6,8,12]$	$[100,10,70]_8$
$[92,10,63]_8$	$[1,2,3,4,5,6,7,8,9,10,12]$	$[103,10,73]_8$

Table 6.5: Some best known codes from puncturing in $\mathbb{F}_8$

7. Improved Codes From Generalised AG Codes

7.1 Introduction

The minimum distance of a code is an important measure of performance in coding theory. It is always desirable to obtain an error correcting code with the maximum possible minimum distance d, given a code length n and code dimension k. In 1981, Goppa (Goppa, 1988) introduced a family of codes with very good properties using principles from algebraic geometry. These codes were later shown to include a class of asymptotically good codes in (Tsfasman et al., 1982) that beat the Gilbert-Varshamov bound for all fields with sizes both square and greater or equal to 49. Algebraic geometry codes and codes obtained from them feature prominently in the databases of best known codes (Grassl, 2007) and (Schimd and Shurer, 2004) for an appreciable range of code lengths for different field sizes q. Generalised algebraic geometry codes were first presented by Niederreiter et al. (1999); Xing et al. (1999b). A subsequent paper by Ozbudak and Stichtenoth (1999) shed more light on the construction. AG codes as defined by Goppa utilised places of degree one or rational places. Generalised AG codes however were constructed by Xing *et al* using places of higher degree (including places of degree one). In (Xing et al., 1999a), the authors presented a method of constructing generalised AG codes which uses a concatenation concept. The paper showed that best known codes were obtainable via this construction. In (Ding et al., 2000) it was shown that the method can be effective in constructing new codes and the authors presented 59 codes in finite fields $\mathbb{F}_4$, $\mathbb{F}_8$ and $\mathbb{F}_9$ better than the codes in (Grassl, 2007). In (Leung et al., 2002), the authors presented a construction method based on (Xing et al., 1999a) that uses a subfield image concept and obtained new binary codes as a result. In (Xing and Yeo, 2007) the authors presented some new curves as well as 129 new codes in $\mathbb{F}_8$ and $\mathbb{F}_9$. In this Chapter we present 237 new improvements to codes defined in $\mathbb{F}_{16}$ in the tables in (Schimd and Shurer, 2004) from a generalised construction of AG codes by Xing et al. (1999a). In addition many improvements to constructible codes in (Schimd and Shurer, 2004) are also presented.

7.2 Concept of Places of Higher Degree

Recall from Chapter 4 that a two dimensional affine space $\mathbb{A}^2(\mathbb{F}_q)$ is given by the set of points

$$\{(\alpha,\beta):\alpha,\beta\in\mathbb{F}_q\}$$

while its projective closure $\mathbb{P}^2(\mathbb{F}_q)$ is given by the set of equivalence points

$$\{\{(\alpha:\beta:1)\}\cup\{(\alpha:1:0)\}\cup\{(1:0:0)\}:\alpha,\beta\in\mathbb{F}_q\}.$$

Given a homogeneous polynomial $F(x,y,z)$, a curve $\mathscr{X}/\mathbb{F}_q$ defined in $\mathbb{P}^2(\mathbb{F}_q)$ is a set of distinct points

$$\mathscr{X}/\mathbb{F}_q=\{T\in\mathbb{P}^2(\mathbb{F}_q):F(T)=0\}$$

Let $\mathbb{F}_{q^\ell}$ be an extension of the field $\mathbb{F}_q$, the Frobenius automorphism is given as

$$\begin{aligned}&\phi_{q,\ell}:\mathbb{F}_{q^\ell}\to\mathbb{F}_{q^\ell}\\&\phi_{q,\ell}(\beta)=\beta^q\qquad\beta\in\mathbb{F}_{q^\ell}\end{aligned}$$

and its action on a projective point $(x:y:z)$ in $\mathbb{F}_{q^\ell}$ is

$$\phi_{q,\ell}((x:y:z))=(x^q:y^q:z^q).$$

7.1 Definition (Place of Degree from (Walker, 2000)). *A place of degree ℓ is a set of ℓ points of a curve defined in the extension field $\mathbb{F}_{q^\ell}$ denoted by $\{T_0,T_1,\ldots,T_{\ell-1}\}$ where each $T_i=\phi_{q,l}^i(T_0)$. Places of degree one are called rational places.*

Example 7.1: Consider the curve in $\mathbb{F}_4$ defined as,

$$F(x,y,z)=x$$

The curve has the following projective rational points (points of degree 1),

$$\begin{aligned}&(0:0:1)\quad(0:1:1)\quad(0:\alpha:1)\quad(0:\alpha^2:1)\\&(0:1:0)\end{aligned}$$

where α is the primitive polynomial of $\mathbb{F}_4$. The curve has the following places of degree 2,

$$\{(0:\beta:1),(0:\beta^4:1)\} \quad \{(0:\beta^2:1),(0:\beta^8:1)\}$$
$$\{(0:\beta^3:1),(0:\beta^{12}:1)\} \quad \{(0:\beta^6:1),(0:\beta^9:1)\}$$
$$\{(0:\beta^7:1),(0:\beta^{13}:1)\} \quad \{(0:\beta^{11}:1),(0:\beta^{14}:1)\}$$

where β is the primitive element of $\mathbb{F}_{16}$.

7.3 Generalised Construction I

This section gives details of the construction of generalised AG codes as described in (Xing et al., 1999a). Two maps that are useful in the construction of generalised AG codes are now described. Observe that $\mathbb{F}_q$ is a subfield of $\mathbb{F}_{q^\ell}$ for all $\ell \geq 2$. It is then possible to map $\mathbb{F}_{q^\ell}$ to an ℓ-dimensional vector space with elements from $\mathbb{F}_q$ using a suitable basis. The map π_ℓ is defined as such,

$$\pi_\ell : \mathbb{F}_{q^\ell} \rightarrow \mathbb{F}_q^\ell$$
$$\pi_\ell(\beta) = (c_1\, c_2 \ldots c_\ell) \qquad \beta \in \mathbb{F}_{q^\ell}\ ,\ c_i \in \mathbb{F}_q.$$

Suppose $(\gamma_1, \gamma_2, \ldots, \gamma_\ell)$ forms a suitable basis of the vector space $\mathbb{F}_q^\ell$, then $\beta = c_1\gamma_1 + c_2\gamma_2 + \cdots + c_\ell\gamma_\ell$. Finally the map $\sigma_{\ell,n}$ is used to represent an encoding map from an ℓ-dimensional message space in $\mathbb{F}_q$ to an n-dimensional code space,

$$\sigma_{\ell,n} : \mathbb{F}_q^\ell \rightarrow \mathbb{F}_q^n$$

with $\ell \leq n$.

A description of generalised AG codes as presented in (Ding et al., 2000; Xing et al., 1999a; Xing and Yeo, 2007) is now presented. Let $F = F(x,y,z)$ be a homogeneous polynomial defined in $\mathbb{F}_q$. Let g be the genus of the curve $\mathscr{X}/\mathbb{F}_q$ corresponding to the polynomial F. Also let $P_1, P_2, \ldots, P_r$ be r distinct places of $\mathscr{X}/\mathbb{F}_q$ and $k_i = deg(P_i)$ (deg is degree of). W is a divisor of the curve $\mathscr{X}/\mathbb{F}_q$ such that $W = P_1 + P_2 + \cdots + P_r$ and G a divisor so that $supp(W) \cap supp(G) = \varnothing$. More specifically $G = m(Q - R)$ where $deg(Q) = deg(R) + 1$ for arbitrary divisors Q and R. Associated with the divisor G is a Riemann-Roch space $\mathscr{L}(G)$ with $m = deg(G)$) an integer, $m \geq 0$. From the Riemann-Roch theorem it is known that the dimension of $\mathscr{L}(G)$ is given by $l(G)$ and

$$l(G) \geq m - g + 1$$

with equality when $m \geq 2g - 1$. Also associated with each P_i is a q-ary code C_i with parameters $[n_i, k_i = deg(P_i), d_i]_q$ with the restriction that $d_i \leq k_i$. Let $\{f_1, f_2, .., f_k : f_l \in \mathscr{L}(G)\}$ denote a set of k linearly independent elements of $\mathscr{L}(G)$ that form a basis. A generator matrix for a generalised AG code is given as such,

#	P_i	$deg(P_i)$
P_1	$(0:1:0)$	1
P_2	$(0:0:1)$	1
P_3	$(1:0:1)$	1
P_4	$(1:1:1)$	1
P_5	$\{(\alpha:1:1),(\alpha^2:1:1)\}$	2
P_6	$\{(\alpha:\alpha+1:1),(\alpha^2:\alpha:1)\}$	2

Table 7.1: Places of $\mathscr{X}/\mathbb{F}_2$

$$M = \begin{bmatrix} \sigma_{k_1,n_1}(\pi_{k_1}(f_1(P_1))) & \cdots\cdots & \sigma_{k_r,n_r}(\pi_{k_r}(f_1(P_r))) \\ \sigma_{k_1,n_1}(\pi_{k_1}(f_2(P_1))) & \cdots\cdots & \sigma_{k_r,n_r}(\pi_{k_r}(f_2(P_r))) \\ \vdots & \ddots & \vdots \\ \sigma_{k_1,n_1}(\pi_{k_1}(f_k(P_1))) & \cdots\cdots & \sigma_{k_r,n_r}(\pi_{k_r}(f_k(P_r))) \end{bmatrix}$$

where $f_l(P_i)$ is an evaluation of a polynomial and basis element f_l at a point P_i, π_{k_i} is a mapping from $\mathbb{F}_{q^{k_i}}$ to $\mathbb{F}_q$ and σ_{k_i,n_i} is the encoding of a message vector in $\mathbb{F}_q^{k_i}$ to a code vector in $\mathbb{F}_q^{n_i}$. It is desirable to choose the maximum possible minimum distance for all codes C_i so that $d_i = k_i$. The same code is used in the map σ_{k_i,n_i} for all points of the same degree k_i i.e. the code C_j has parameters $[n_j, j, d_j]_q$ for a place of degree j. Let A_j be an integer denoting the number of places of degree j and B_j be an integer such that $0 \le B_j \le A_j$. If t is the maximum degree of any place P_i that is chosen in the construction, then the generalised AG code is represented as a $C_1(k;t;B_1,B_2,\ldots,B_t;d_1,d_2,\ldots,d_t)$. Let $[n,k,d]_q$ represent a linear code in $\mathbb{F}_q$ with length n, dimension k and minimum distance d, then a generalised AG code is given by the parameters (Xing et al., 1999a),

$$\begin{aligned} k &= l(G) \ge m - g + 1 \\ n &= \sum_{i=1}^{r} n_i = \sum_{j=1}^{t} B_j n_j \\ d &\ge \sum_{i=1}^{r} d_i - g - k + 1 = \sum_{j=1}^{t} B_j d_j - g - k + 1. \end{aligned}$$

Example 7.2: Let $F(x,y,z) = x^3 + xyz + xz^2 + y^2z$ (Xing et al., 1999a) be a polynomial in $\mathbb{F}_2$. The curve $\mathscr{X}/\mathbb{F}_2$ has genus $g = 1$ and $A_1 = 4$ places of degree 1 and $A_2 = 2$ places of degree 2. Table 7.1 gives the places of $\mathscr{X}/\mathbb{F}_2$ up to degree 2. The field $\mathbb{F}_{2^2}$ is defined by a primitive polynomial $s^2 + s + 1$ with α as its primitive element. Points $R = (1 : a^3 + a^2 : 1)$ as a place of degree 4 and $Q = (1 : b^4 + b^3 + b^2 : 1)$ as a place of degree 5 are also chosen while a and b are primitive elements of $\mathbb{F}_{2^4}$ (defined by the polynomial $s^4 + s^3 + s^2 + s + 1$) and $\mathbb{F}_{2^5}$ (defined by the polynomial $s^5 + s^2 + 1$) respectively. The divisor W is $W = P_1 + \cdots + P_6$. The basis of the Riemann-Roch

space $\mathscr{L}(2D)$ with $D = Q - R$ and $m = 2$ is obtained with computer algebra software MAGMA (Bosma et al., 1997) as,

$$\begin{aligned} f_1 &= (x^7 + x^3 + x)/(x^{10} + x^4 + 1)y \\ &\quad + (x^{10} + x^9 + x^7 + x^6 + x^5 + x + 1)/(x^{10} + x^4 + 1) \\ f_2 &= (x^8 + x^7 + x^4 + x^3 + x + 1)/(x^{10} + x^4 + 1)y \\ &\quad + (x^8 + x^4 + x^2)/(x^{10} + x^4 + 1) \end{aligned}$$

For the map σ_{k_i,n_i} the codes; c_1 a $[1,1,1]_2$ cyclic code for places of degree 1 and c_2 a $[3,2,2]_2$ cyclic code for places of degree 2 are used. For the map π_2 which applies to places of degree 2 a polynomial basis $[\gamma_1, \gamma_2] = [1, \alpha]$ is used. Only the first point in the place P_i for $deg(P_i) = 2$ in the evaluation of f_1 and f_2 at P_i is utilised. The generator matrix M of the resulting $[10,2,6]_2$ generalised AG code over $\mathbb{F}_2$ is,

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

Example 7.3: Consider again the polynomial $F(x,y,z) = x^3 + xyz + xz^2 + y^2z$ with coefficients from $\mathbb{F}_2$ whose curve (with genus equal to 1) has places up to degree 2 as in Table 7.1. An element f of the Riemann Roch space defined by the divisor $G = (R - Q)$ with $Q = (a : a^3 + a^2 : 1)$ and $R = (b : b^4 + b^3 + b^2 + b + 1 : 1)$ where a and b primitive elements of $\mathbb{F}_{2^4}$ and $\mathbb{F}_{2^5}$ (since the curve has no place of degree 3) respectively, is given by,

$$\begin{aligned} f &= (x^3x + x^2z^2 + z^4)y/(x^5 + x^3z^2 + z^5) \\ &\quad + (x^5 + x^4z + x^3z^2 + z^3x^2 + xz^4 + z^5)/(x^5 + x^3z^2 + z^5) \end{aligned}$$

Evaluating f at all the 5 places $\mathscr{P}_i$ from the Table 7.1 and using the map $\pi_{\deg(P_i)}$ that maps all evaluations to $\mathbb{F}_2$ results in,

$$[\ \overbrace{1 \mid 1 \mid 0 \mid 1}^{f(\mathscr{P}_i)\,|_{\deg(\mathscr{P}_i)=1}} \quad \underbrace{1 \mid \alpha^2}_{f(\mathscr{P}_i)\,|_{\deg(\mathscr{P}_i)=2}} \ ]$$

This forms the code $[6,1,5]_4$ [1]. In $\mathbb{F}_2$ this becomes,

$$[\,1 \mid 1 \mid 0 \mid 1 \mid \underbrace{1\ 0}_{1} \mid \underbrace{1\ 1}_{\alpha^2}]$$

[1] From Bezout's $d_{min} = n - m = n - k - g + 1$

which forms the code $[8,1,5]_2$. Short auxiliary codes $[1,1,1]_2$ to encode $f(\mathscr{P}_i)\,|_{\deg(\mathscr{P}_i)=1}$ and $[3,2,2]_2$ to encode $f(\mathscr{P}_i)\,|_{\deg(\mathscr{P}_i)=2}$ are used. The resulting codeword of a generalised AG code is ,

$$[\ 1\ |\ 1\ |\ 0\ |\ 1\ |\ 1\ \ 0\ \ 1\ |\ 1\ \ 1\ \ 0\ ].$$

This forms the code $[10,1,7]_2$.

7.3.1 Results

Four polynomials and their associated curves are used to obtain codes in $\mathbb{F}_{16}$ better than the best known codes in (Schimd and Shurer, 2004). The four polynomials are given in Table 7.2 while Table 7.3 gives a summary of the properties of their associated curves (with $t = 4$). w is the primitive element of $\mathbb{F}_{16}$. The number of places of degree j, A_j, is determined by computer algebra system MAGMA (Bosma et al., 1997). The best known linear codes from (Schimd and Shurer, 2004) over $\mathbb{F}_{16}$ with $j = d_j$ for $1 \le j \le 4$ are

$$[1,1,1]_{16}\quad [3,2,2]_{16}\quad [5,3,3]_{16}\quad [7,4,4]_{16}$$

which correspond to C_1, C_2, C_3 and C_4 respectively. Since $t = 4$ for all the codes in this paper and

$$[d_1,d_2,d_3,d_4] = [1,2,3,4]$$

The representation $C_1(k;t;B_1,B_2,\ldots,B_t;d_1,d_2,\ldots,d_t)$ is shortened as such,

$$C_1(k;t;B_1,B_2,\ldots,B_t;d_1,d_2,\ldots,d_t) \equiv C_1(k;B_1,B_2,\ldots,B_t).$$

Tables 7.4-7.5 gives new codes that improve on both constructible codes in (Schimd and Shurer, 2004). Tables 7.6-7.7 show new codes with better minimum distance than codes in (Schimd and Shurer, 2004). It is also worth noting that codes of the form $C_1(k;N,0,0,0)$ are simply Goppa codes (defined with only rational points). The symbol # in the Tables 7.4-7.7 denotes the number of new codes from each generalised AG code $C_1(k;B_1,B_2,\ldots,B_t)$. The tables in (Geer et al., 2009) contain curves known to have the most number of rational points for a given genus. Over $\mathbb{F}_{16}$ the curve with the highest number of points with genus $g = 12$ from (Geer et al., 2009) has 88 rational points, was constructed using class field theory and is not defined by an explicit polynomial. On the other hand the curve $\mathscr{X}_1/\mathbb{F}_{16}$ obtained by Kummer covering of the projective line in (Shabat, 2001) has $A_1 = 83$ rational points and genus $g = 12$ and is explicitly presented. Codes from this curve represent the best constructible codes in $\mathbb{F}_{16}$ with code length 83. The curve $\mathscr{X}_2/\mathbb{F}_{16}$ is defined by the well-known Hermitian polynomial.

Tables 7.4-7.8 give the new codes obtained from $\mathscr{X}_2/\mathbb{F}_{16}$,$\mathscr{X}_3/\mathbb{F}_{16}$ and $\mathscr{X}_4/\mathbb{F}_{16}$.

$F_1 = x^5z^{10} + x^3z^{12} + xz^{14} + y^{15}$
$F_2 = x^5 + y^4z + yz^4$
$F_3 = x^{16} + x^4y^{15} + x^4 + xy^{15} + w^4y^{15} + w^4$
$F_4 = x^{28} + wx^{20} + x^{18} + w^{10}x^{17} + w^{10}x^{15} + w^4x^{14} + w^3x^{13} + w^3x^{12} + wx^{11} + x^{10} + w^{11}x^9 + w^{12}x^8 + w^{14}x^7 + w^{13}x^6y^2 + w^9x^6y + w^6x^6 + w^2x^5y^2 + w^{13}x^5y + w^{14}x^5 + w^{14}x^4y^4 + w^7x^4y^2 + w^6x^4y + w^9x^4 + w^8x^3y^4 + w^{11}x^3y + w^4x^3 + w^{11}x^2y^4 + w^{11}x^2y^2 + wx^2y + w^5x^2 + w^8xy^4 + w^6xy^2 + w^9xy + w^{11}y^8 + y^4 + w^2y^2 + w^3y$

Table 7.2: Polynomials in $\mathbb{F}_{16}$

Curve	Genus	A_1	A_2	A_3	A_4	Reference
$\mathcal{X}_1$	12	83	60	1320	16140	(Shabat, 2001)
$\mathcal{X}_2$	6	65	0	1600	15600	Hermitian curve
$\mathcal{X}_3$	40	225	0	904	16920	(Garcia and Quoos, 2001)
$\mathcal{X}_4$	13	97	16	1376	15840	(Geer and Vlugt, 2000) via (Grassl, 2010)

Table 7.3: Properties of $\mathcal{X}_i/\mathbb{F}_{16}$

Codes	k Range	Description	#
$[83,k,d \geq 72-k]_{16}$	$8 \leq k \leq 52$	$C_1(k;[83,0,0,0])$	45
$[89,k,d \geq 76-k]_{16}$	$9 \leq k \leq 54$	$C_1(k;[83,2,0,0])$	46
$[94,k,d \geq 79-k]_{16}$	$10 \leq k \leq 57$	$C_1(k;[83,2,1,0])$	48
$[92,k,d \geq 78-k]_{16}$	$9 \leq k \leq 57$	$C_1(k;[83,3,0,0])$	49
$[98,k,d \geq 82-k]_{16}$	$11 \leq k \leq 59$	$C_1(k;[83,5,0,0])$	49

Table 7.4: Best Constructible Codes from $\mathcal{X}_1/\mathbb{F}_{16}$

Codes	k Range	Description	#
$[72,k,d \geq 64-k]_{16}$	$11 \leq k \leq 50$	$C_1(k;[65,0,0,1])$	40
$[79,k,d \geq 68-k]_{16}$	$11 \leq k \leq 48$	$C_1(k;[65,0,0,2])$	38
$[77,k,d \geq 67-k]_{16}$	$10 \leq k \leq 51$	$C_1(k;[65,0,1,1])$	42
$[75,k,d \geq 66-k]_{16}$	$9 \leq k \leq 51$	$C_1(k;[65,0,2,0])$	43

Table 7.5: Best Constructible Codes from $\mathcal{X}_2/\mathbb{F}_{16}$

Codes	k Range	Description	#
$[70,k,d \geq 63-k]_{16}$	$10 \leq k \leq 50$	$C_1(k;[65,0,1,0])$	41

Table 7.6: New Codes from $\mathcal{X}_2/\mathbb{F}_{16}$

Code	k Range	Description	#
$[232,k,190-k]$	$102 \geq k \geq 129$	$C_1(k;[225,0,0,1])$	28
$[230,k,189-k]$	$100 \geq k \geq 129$	$C_1(k;[225,0,1,0])$	30
$[235,k,192-k]$	$105 \geq k \geq 121$	$C_1(k;[225,0,2,0])$	17

Table 7.7: New Codes from $\mathcal{X}_3/\mathbb{F}_{16}$

Codes	k Range	Description	#
$[102,k,88-k]$	$8 \le k \le 66$	$C(k;[97,0,1,0])$	59
$[103,k,89-k]$	$8 \le k \le 68$	$C(k;[97,2,0,0])$	61
$[106,k,91-k]$	$k = 8$	$C(k;[97,3,0,0])$	1

Table 7.8: New Codes from $\mathscr{X}_4/\mathbb{F}_{16}$

7.4 Generalised Construction II

This section describes the construction of generalised AG codes as described in (Leung et al., 2002). This method is a variation of the first method in (Xing et al., 1999a) and codes constructed from this construction can be seen as subfield image codes. Suppose $\mathbb{F}_q \subset \mathbb{F}_{q^e} \subset \mathbb{F}_{q^{e\ell}}$ with $\ell \ge 2$ and $e > 1$. It is then possible to map $\mathbb{F}_{q^{e\ell}}$ to an $e\ell \times e\ell$- matrix with elements from $\mathbb{F}_q$ (see Section 2.3.3.1). The map $\psi_{e\ell}$ is defined as such,

$$\psi_{e\ell} : \mathbb{F}_{q^{e\ell}} \to \mathbb{F}_q^{e\ell \times e\ell}$$

The map $\psi_{e\ell}$ acts on a vector $a = (a_1, \ldots, a_u)$ as such,

$$\psi_{e\ell}((a_1, \ldots, a_u)) = \psi_{e\ell}(a_1)|\ldots|\psi_{e\ell}(a_u) \quad a_i \in \mathbb{F}_{q^{e\ell}}$$

where | denotes matrix concatenation. The map $\sigma_{e\ell,n}$ is also defined,

$$\sigma_{e\ell,n} : \mathbb{F}_q^{e\ell \times e\ell} \to \mathbb{F}_q^{e\ell \times n}$$
$$\sigma_{e\ell,n}(M) = M \times G$$

where M is matrix with dimension $e\ell \times e\ell$ and G is the generator matrix of a code with dimension $e\ell$ and length n. Let $F = F(x,y,z)$ be a homogeneous polynomial defined in $\mathbb{F}_{q^e}$. Let g be the genus of the curve $\mathscr{X}/\mathbb{F}_{q^e}$ corresponding to the polynomial F. Also let $P_1, P_2, \ldots, P_r$ be r distinct places of $\mathscr{X}/\mathbb{F}_{q^e}$ and $k_i = deg(P_i)$ (deg is degree of). W is a divisor of the curve $\mathscr{X}/\mathbb{F}_{q^e}$ such that $W = P_1 + P_2 + \cdots + P_r$ and G a divisor so that

$$supp(W) \cap supp(G) = \varnothing.$$

More specifically $G = m(Q - R)$ where $deg(Q) = deg(R) + 1$. Associated with the divisor G is a Riemann-Roch space $\mathscr{L}(G)$ with $m = deg(G))$ an integer, $m \ge 0$. From the Riemann-Roch theorem it is known that the dimension of $\mathscr{L}(G)$ is given by $l(G)$ and

$$l(G) \ge m - g + 1$$

with equality when $m \ge 2g - 1$. Also associated with each P_i is a q-ary code C_i with parameters $[n_i, ek_i = e \cdot deg(P_i), d_i]_q$ with the restriction that $d_i \le ek_i$. Let $\{f_1, f_2, .., f_k : f_l \in \mathscr{L}(G)\}$ denote a set of k linearly independent elements of $\mathscr{L}(G)$

that form a basis. A generator matrix for a generalised AG code is given as such,

$$M = \begin{bmatrix} \sigma_{ek_1,n_1}(\psi_{ek_1}(\pi_{k_1}(f_1(P_1)))) & \cdots\cdots & \sigma_{ek_r,n_r}(\psi_{ek_r}(\pi_{k_r}(f_1(P_r)))) \\ \sigma_{ek_1,n_1}(\psi_{ek_1}(\pi_{k_1}(f_2(P_1)))) & \cdots\cdots & \sigma_{ek_r,n_r}(\psi_{ek_r}(\pi_{k_r}(f_2(P_r)))) \\ \vdots & \ddots & \vdots \\ \sigma_{ek_1,n_1}(\psi_{ek_1}(\pi_{k_1}(f_k(P_1)))) & \cdots\cdots & \sigma_{ek_r,n_r}(\psi_{ek_r}(\pi_{k_r}(f_k(P_r)))) \end{bmatrix}$$

It is desirable to choose the maximum possible minimum distance for all codes C_i so that $d_i = ek_i$. The same code is used in the map σ_{ek_i,n_i} for all points of the same degree k_i i.e. the code C_j has parameters $[n_j, ej, d_j]_q$ for a place of degree j. Let A_j be an integer denoting the number of places of degree j and B_j be an integer such that $0 \leq B_j \leq A_j$. If t is the maximum degree of any place P_i chosen in the construction, then the generalised AG code is represented as a $C_2(k;t;B_1,B_2,\ldots,B_t;d_1,d_2,\ldots,d_t)$. Let $[n,k,d]_q$ represent a linear code in $\mathbb{F}_q$ with length n, dimension k and minimum distance d, then a generalised type II AG code is given by the parameters (Leung et al., 2002),

$$\begin{aligned} k &= e \cdot l(G) \geq e(m-g+1) \\ n &= \sum_{i=1}^{r} n_i = \sum_{j=1}^{t} B_j n_j \\ d &\geq \sum_{i=1}^{r} d_i - e \cdot deg(G) = \sum_{i=1}^{r} d_i - k \end{aligned}$$

No improvements were obtained using this method however many best known codes have been obtained.

7.5 Summary

The concept of place of higher degrees of curves was presented. This notion was used in the construction of two types of generalised AG codes. Using the generalised Construction I 237 improvements to the tables in (Schimd and Shurer, 2004) were found. Further results are also obtainable from trivial modifications on the new codes like shortening, padding and puncturing. In addition many improvements on constructible codes in the table in (Schimd and Shurer, 2004) are presented. Finding curves with many places of small degree and small genera will result in many new codes using these methods. Tables in (Schimd and Shurer, 2004) can be improved via these methods using curves with many points in large finite fields.

8. Improved Codes From Goppa Codes

8.1 Introduction

Goppa introduced a class of linear codes in (Goppa, 1972) and (Goppa, 1971) commonly referred to as *Goppa codes* or $\Gamma(L,G)$ codes. These codes form an important subclass of alternant codes and meet the famous Gilbert-Varshamov bound. $\Gamma(L,G)$ codes have good properties and some of these codes have the best known minimum distance of any known codes with the same length and rate. The codes are mainly defined in a finite field $\mathbb{F}_q$ and are sub-field sub-codes of generalised Reed Solomon codes defined in an extension field of $\mathbb{F}_q$. Goppa in a subsequent paper (Goppa, 1972) showed several methods of extending the length of $\Gamma(L,G)$ codes. Similarly Sugiyama *et al* (Sugiyama et al., 1976) presented binary codes derived from $\Gamma(L,G)$ codes by extending their length and produced some good codes as a result. In this chapter a construction of extended nonbinary Goppa codes and some improved codes that have better minimum distance than the best known codes in the tables from (Grassl, 2007) with the same length and dimension are presented. This construction is a generalisation of the method in (Sugiyama et al., 1976) for binary Goppa codes.

Section 8.2 gives a brief background on Goppa codes and a definition that suits the purposes of this chapter. Section 8.3 gives a generalisation of Construction P (Sugiyama et al., 1976) for binary codes and establishes the parameters of nonbinary codes obtained therefrom. Section 8.4 shows that certain extended Goppa Codes can be seen as BCH codes and an instance of the construction from Section 8.3 is used to present improved codes. And finally, Section 8.6 gives a summary of the codes found using the construction method and further results from nested codes using construction X.

8.2 Goppa Codes

Recall the description of Goppa codes from Section 3.4. In designing Goppa codes, it is usually desirable to obtain codes as long as possible and hence the Goppa polynomial $G(x)$ is commonly chosen to have no roots in the field $\mathbb{F}_{q^m}$, in which case the length of the code is equal to the size of the field i.e. $n = q^m$. For our purposes

we are interested in Goppa codes whose polynomial has roots in the field $\mathbb{F}_{q^m}$. A useful relationship between the parity check matrix of a Goppa code (defined with a polynomial with roots in its coefficient field) and the Cauchy matrix was presented in (Sugiyama et al., 1976) and more explicitly in (Tzeng and Zimmermann, 1975).

8.1 *Theorem (From* (Tzeng and Zimmermann, 1975, Appendix)*).* *A $\Gamma(L,G)$ defined by a polynomial $G(x) = \prod_{\mu=1}^{\ell}(x-\beta_\mu)^{r_\mu}$ with each β_μ distinct, $r_\mu > 0$ and $\beta_\mu \in \mathbb{F}_{q^m}$ satisfies the parity equations*

$$\sum_{i=0}^{n-1} \frac{c_i}{(\beta_\mu - \alpha_i)^j} = 0 \quad \text{for } j = 1,\ldots,r_\mu,\ \mu = 1,\ldots,\ell.$$

The parity check matrix of the code can be expressed as

$$H = \begin{bmatrix} H_{r_1} \\ H_{r_2} \\ \vdots \\ H_{r_\ell} \end{bmatrix}, \tag{8.1}$$

where

$$H_{r_\mu} = \begin{bmatrix} \frac{1}{(\beta_\mu-\alpha_0)} & \frac{1}{(\beta_\mu-\alpha_1)} & \cdots & \frac{1}{(\beta_\mu-\alpha_{n-1})} \\ \frac{1}{(\beta_\mu-\alpha_0)^2} & \frac{1}{(\beta_\mu-\alpha_1)^2} & \cdots & \frac{1}{(\beta_\mu-\alpha_{n-1})^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{(\beta_\mu-\alpha_0)^{r_\mu}} & \frac{1}{(\beta_\mu-\alpha_1)^{r_\mu}} & \cdots & \frac{1}{(\beta_\mu-\alpha_{n-1})^{r_\mu}} \end{bmatrix}. \tag{8.2}$$

This code with symbols in $\mathbb{F}_q$ and defining set $L = \mathbb{F}_{q^m} \setminus \{\beta_1,\ldots,\beta_\ell\} = \{\alpha_0,\ldots,\alpha_{n-1}\}$ has parameters

$$\begin{array}{ll} \text{length:} & n = |L|, \\ \text{redundancy:} & n-k \le m\left(\sum_{\mu=1}^{\ell} r_\mu\right), \\ \text{distance:} & d \ge \sum_{\mu=1}^{\ell} r_\mu + 1. \end{array}$$

A special case is when $r_\mu = 1$ for all μ when the parity matrix of the $\Gamma(L,G)$ code becomes

$$H = \begin{bmatrix} \frac{1}{\beta_1-\alpha_0} & \frac{1}{\beta_1-\alpha_1} & \cdots & \frac{1}{\beta_1-\alpha_{n-1}} \\ \frac{1}{\beta_2-\alpha_0} & \frac{1}{\beta_2-\alpha_1} & \cdots & \frac{1}{\beta_2-\alpha_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\beta_\ell-\alpha_0} & \frac{1}{\beta_\ell-\alpha_1} & \cdots & \frac{1}{\beta_\ell-\alpha_{n-1}} \end{bmatrix},$$

which is equivalent to a Cauchy matrix. It is also the parity check matrix of a separable Goppa code.

8.2.1 Modified Goppa Codes

In (Goppa, 1971), Goppa defined modified Goppa codes. He showed that adding a row of all 1's to the parity check matrix of a Goppa code increases both the minimum distance and the redundancy by one.

8.2 *Theorem (From* (Goppa, 1971, Theorem 3)*).* *A modified Goppa code $\tilde{\mathscr{C}}$ with the parity check matrix*

$$\tilde{H} = \begin{bmatrix} 11\ldots1 \\ H \end{bmatrix},$$

where H is the parity check matrix of a $\Gamma(L,G)$ code with a Goppa polynomial $G(x)$ with degree r defined with coefficients in $\mathbb{F}_{q^m}$, has parameters

$$\begin{aligned} &\textit{length:} && n = |L|, \\ &\textit{redundancy:} && n-k \le mr+1, \\ &\textit{distance:} && d \ge r+2. \end{aligned}$$

The use of modified Goppa codes is most effective when the codes have symbols in the field $\mathbb{F}_q$ for which $q \neq 2$. It is possible to extend a modified Goppa code by adding a parity check on the row with all 1's

$$\tilde{H}_e = \begin{bmatrix} 11\ldots1 & 1 \\ H & 0 \end{bmatrix}. \tag{8.3}$$

This extended and modified Goppa code has parameters

$$\begin{aligned} &\text{length:} && n' = |L|+1, \\ &\text{redundancy:} && n'-k' \le mr+1. \end{aligned}$$

8.3 *Theorem.* *The minimum distance of an extended and modified Goppa code defined with a polynomial $G(x)$ of degree r is lower bounded by $d' \ge r+2$.*

Proof. Let $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ be a non-zero codeword of the Goppa code $\mathscr{C}$ defined by $G(x)$ of degree r and the parity check matrix H in (8.3). A codeword of an extended and modified Goppa code is then of the form

$$\mathbf{c}_e = (c_0, c_1, \ldots, c_{n-1}, -\sum_{i=0}^{n-1} c_i).$$

If $c_n = -\sum_{i=0}^{n-1} c_i = 0$, then $\mathbf{c}$ is a codeword of the modified Goppa code of Theo-

rem 8.2 and its weight is at least $r+2$. Otherwise, $c_n \neq 0$ and hence

$$\mathrm{wgt}(\mathbf{c}_e) = \mathrm{wgt}(\mathbf{c}) + 1 \geq r+2. \qquad \blacksquare$$

In the literature, these extended and modified Goppa codes are simply called extended Goppa codes (MacWilliams and Sloane, 1983).

8.3 Code Construction

The construction presented below is a generalisation of Construction P in (Sugiyama et al., 1976) from binary to nonbinary codes.

We start with extended and modified Goppa codes defined in the previous section and a Goppa polynomial with roots exclusively in $\mathbb{F}_{q^m}$. Consider the Goppa polynomial

$$G(x) = \prod_{\mu=1}^{\ell} (x-\beta_\mu)^{r_\mu} \tag{8.4}$$

of degree $r = \sum_{\mu=1}^{\ell} r_\mu$ with distinct roots $\beta_\mu \in \mathbb{F}_{q^m}$ and $r_\mu > 0$. The codes $\mathscr{C}_{\mathrm{P}}$ are defined via with parity check matrix

$$H_{\mathrm{P}} = \begin{bmatrix} 11\ldots1 & 1 & 0 & 0 & \cdots & 0 \\ H_{r_1} & 0 & H_{\mathrm{I}_1} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{r_\ell} & 0 & 0 & 0 & \cdots & H_{\mathrm{I}_\ell} \end{bmatrix}. \tag{8.5}$$

The first $q^m - \ell + 1$ columns of H_{P} contain the parity check matrix $\tilde{H}_e$ of the extended and modified Goppa code given in (8.3), where the matrices H_{r_μ} are defined in (8.2). For each of the matrices H_{r_μ}, we add an $r_\mu \times m$ matrix H_{I_μ} of the form

$$H_{\mathrm{I}_\mu} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \\ 1 & \alpha & \alpha^2 & \cdots & \alpha^{m-1} \end{bmatrix},$$

where α is a primitive element of the field $\mathbb{F}_{q^m}$. Clearly the code $\mathscr{C}_{\mathrm{P}}$ has length and redundancy

$$\begin{aligned} &\text{length:} && n = q^m - \ell + m\ell + 1, \\ &\text{redundancy:} && n - k \leq mr + 1. \end{aligned}$$

To obtain a lower bound on the minimum distance of the codes $\mathscr{C}_{\text{P}}$, we can basically follow the logic and presentation of the proof of Theorem 7 in (Sugiyama et al., 1976).

8.4 *Theorem.* *The minimum distance of the code $\mathscr{C}_{\text{P}}$ is lower bounded by $d \geq r+2$.*

Proof. Let $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell)$ be a codeword of $\mathscr{C}_{\text{P}}$, where

$$\mathbf{c}_0 = (a_1, a_2, \ldots, a_{q^m-\ell}, a_{q^m-\ell+1})$$

is a codeword of a modified and extended Goppa code $\tilde{\mathscr{C}}_e$ with Goppa polynomial given in Equation (8.4), and $\mathbf{c}_\mu \in \mathbb{F}_q^m$ for $1 \leq \mu \leq \ell$. If at least one of these vectors $\mathbf{c}_\mu$ for $1 \leq \mu \leq \ell$ is non-zero, then $\mathbf{c}_0$ must be non-zero as well since the columns of the submatrices H_{I_μ} are linearly independent over $\mathbb{F}_q$.

Therefore, assume that $\mathbf{c}_0$ is non-zero. Furthermore, let U_{Z} and U_{N} be the sets of integers μ such that $\mathbf{c}_\mu$, $\mu \geq 1$ is zero or non-zero, respectively. For $\mu \in U_{\text{N}}$, by definition $\mathbf{c}_\mu \neq \mathbf{0}$, and $\mathbf{c}_\mu$ has weight at least 1. Hence the weight of $\mathbf{c}$ is lower bounded by $\text{wgt}(\mathbf{c}_0) + |U_{\text{N}}|$.

In order to obtain a bound on the weight of $\mathbf{c}_0$ first note that $\mathbf{c}_\mu \neq \mathbf{0}$ implies that the parity check given by the last row of H_μ in H_{P} does not hold for $\mathbf{c}_0$, but the other parity check equations are fulfilled. Hence $\mathbf{c}_0$ is a codeword of the extended and modified Goppa code with Goppa polynomial

$$\tilde{G}(x) = \prod_{\mu \in U_{\text{N}}} (x-\beta_\mu)^{r_\mu - 1} \prod_{\mu \in U_{\text{Z}}} (x-\beta_\mu)^{r_\mu}.$$

The degree of $\tilde{G}(x)$ is $r - |U_{\text{N}}|$, and hence by Theorem 8.3 $\text{wgt}(\mathbf{c}_0) \geq r - |U_{\text{N}}| + 2$. In summary we get $\text{wgt}(\mathbf{c}) \geq \text{wgt}(\mathbf{c}_0) + |U_{\text{N}}| \geq r + 2$. ∎

An alternative view of the codes $\mathscr{C}_{\text{P}}$ is that each codeword $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_\ell) \in \mathscr{C}_{\text{P}}$ consists of a vector $\mathbf{c}_0$ over the field $\mathbb{F}_q$, while $\mathbf{c}_\mu$, $1 \leq \mu \leq \ell$ are elements of the extension field $\mathbb{F}_{q^m}$ which are mapped to m symbols in $\mathbb{F}_q$ using a basis $(1, \alpha, \ldots, \alpha^{m-1})$.

8.4 $\mathscr{C}_{\text{P}}$ As Extended BCH Codes

A subset of the codes $\mathscr{C}_{\text{P}}$ can be seen as extended BCH codes in which case a better lower bound on the dimension of the codes can be obtained. In this case the Goppa polynomial in (8.4) is defined with $r_1 > 1$ and $r_\mu = 1$ for $2 \leq \mu \leq \ell$, i.e.,

$$G(x) = (x-\beta_1)^{r_1} \prod_{\mu=2}^{\ell} (x-\beta_\mu). \tag{8.6}$$

8.5 ***Theorem (see*** (MacWilliams and Sloane, 1983, Ch. 12. §3, Problem (6))***).*** *A Goppa code $\Gamma(L,G)$ defined with the polynomial $G(x)=(x-\beta)^r$, $\beta\in\mathbb{F}_{q^m}$, and the set $L=\mathbb{F}_{q^m}\setminus\{\beta\}$ corresponds to a BCH code defined in $\mathbb{F}_q$ with length $n=q^m-1$.*

For simplicity we choose $\beta=0$. Then the BCH code $\Gamma(L,x^r)$ has consecutive roots $\{\alpha^{-1},\alpha^{-2},\ldots,\alpha^{-r}\}$ in $\mathbb{F}_{q^m}$. If we choose $\beta_1=0$ in Equation (8.6), the parity check matrix of the modified Goppa code defined with the polynomial in (8.6) and location set $L=\{\alpha_0,\alpha_1,\ldots,\alpha_{n-1}\}$ is given by

$$\tilde{H}=\begin{bmatrix}11\ldots1\\ H_{r_1}\\ H_{r_2}\\ \vdots\\ H_{r_\ell}\end{bmatrix}=\begin{bmatrix}1 & 1 & \cdots & 1\\ \frac{1}{\alpha_0} & \frac{1}{\alpha_1} & \cdots & \frac{1}{\alpha_{n-1}}\\ \frac{1}{\alpha_0^2} & \frac{1}{\alpha_1^2} & \cdots & \frac{1}{\alpha_{n-1}^2}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{1}{\alpha_0^{r_1}} & \frac{1}{\alpha_1^{r_1}} & \cdots & \frac{1}{\alpha_{n-1}^{r_1}}\\ \frac{1}{\beta_2-\alpha_0} & \frac{1}{\beta_2-\alpha_1} & \cdots & \frac{1}{\beta_2-\alpha_{n-1}}\\ \frac{1}{\beta_3-\alpha_0} & \frac{1}{\beta_3-\alpha_1} & \cdots & \frac{1}{\beta_3-\alpha_{n-1}}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{1}{\beta_\ell-\alpha_0} & \frac{1}{\beta_\ell-\alpha_1} & \cdots & \frac{1}{\beta_\ell-\alpha_{n-1}}\end{bmatrix}. \tag{8.7}$$

Suppose $\mathbf{c}_0=(a_1,a_2,\ldots,a_{q^m-\ell})$ is a codeword of the Goppa code $\Gamma(L,x^{r_1-1})$ with $L=\{\beta\in\mathbb{F}_{q^m}:G(\beta)\neq0\}$ corresponding to a shortened BCH code[1] with roots $\{\alpha^{-1},\ldots,\alpha^{-r_1+1}\}$ in $\mathbb{F}_{q^m}$. As noted at the end of the previous section, we can represent $\mathbf{c}\in\mathscr{C}_{\mathrm{P}}$ in the form

$$\mathbf{c}=(a_1,\ldots,a_{q^m-\ell},a_{q^m-\ell+1},\mathbf{c}_1,\mathbf{c}_2,\ldots,\mathbf{c}_\ell), \tag{8.8}$$

where $a_j\in\mathbb{F}_q$ and[2] $\mathbf{c}_\mu\in\mathbb{F}_{q^m}$. We have

$$\begin{aligned}a_{q^m-\ell+1}&=-\sum_{i=1}^{q^m-\ell}a_i,\\ \mathbf{c}_1&=-\sum_{i=1}^{q^m-\ell}a_i\alpha_i^{-r_1},\\ \text{and}\quad \mathbf{c}_\mu&=-\sum_{i=1}^{q^m-\ell}\frac{a_i}{\beta_\mu-\alpha_i}\quad\text{for }\mu>1.\end{aligned}$$

Hence the codes $\mathscr{C}_{\mathrm{P}}$ defined with $G(x)$ in (8.6) can be seen as extending the BCH code $\Gamma(L,x^{r_1-1})$. In many cases the redundancy of the codes is smaller than what is predicted by the bound Section 8.3. Suppose r_{BCH} is the redundancy of the BCH

[1] This shortened BCH code has $n=q^m-\ell$, $n-k\le m(r_1-1)$, and $d\ge r_1$

[2] $\mathbf{c}_\mu$ maps to a vector in $\mathbb{F}_q^m$ using the basis $(1,\alpha,\ldots,\alpha^{m-1})$.

code $\Gamma(L, x^{r_1-1})$ then the parameters of the code $\mathscr{C}_{\text{P}}$ defined by the Goppa polynomial in Equation (8.6) are

$$\begin{aligned} &\text{length:} && n = |L| + m\ell + 1, \\ &\text{redundancy:} && n - k \le m\ell + r_{\text{BCH}} + 1, \\ &\text{distance:} && d \ge r_1 + \ell + 1 = r + 2. \end{aligned}$$

8.4.1 Example

We use as an illustration of the construction a polynomial $G(x) = x^2(x+1)(x+\alpha)(x+\alpha^2)$ with coefficients from $\mathbb{F}_{16}$ to define an extended Goppa code in $\mathbb{F}_4$. The finite field $\mathbb{F}_{16}$ is defined with the primitive polynomial $s^4 + s + 1$ and has α as a primitive element. The set L corresponding to $G(x)$ is then given by

$$L = \mathbb{F}_{16} \setminus \{0, 1, \alpha, \alpha^2\}, \qquad |L| = 12.$$

From (8.7) the parity matrix $\tilde{H}$ of the modified Goppa code over $\mathbb{F}_{16}$ is given as

$$\tilde{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha^{12} & \alpha^{11} & \alpha^{10} & \alpha^9 & \alpha^8 & \alpha^7 & \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 \\ \alpha^9 & \alpha^7 & \alpha^5 & \alpha^3 & \alpha^1 & \alpha^{14} & \alpha^{12} & \alpha^{10} & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^1 & \alpha^{14} & \alpha^5 & \alpha^2 & \alpha^6 & \alpha^{13} & \alpha^8 & \alpha^{10} & \alpha^3 & \alpha^4 & \alpha^9 & \alpha^{12} \\ \alpha^6 & \alpha^0 & \alpha^{13} & \alpha^4 & \alpha^1 & \alpha^5 & \alpha^{12} & \alpha^7 & \alpha^9 & \alpha^2 & \alpha^3 & \alpha^8 \\ \alpha^9 & \alpha^5 & \alpha^{14} & \alpha^{12} & \alpha^3 & \alpha^0 & \alpha^4 & \alpha^{11} & \alpha^6 & \alpha^8 & \alpha^1 & \alpha^2 \end{bmatrix}.$$

$\mathscr{C}_{\text{P}}$ is defined by the parity check matrix H_{P} over $\mathbb{F}_4$, given by

$$H_{\text{P}} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & \overline{\omega} & \omega & \overline{\omega} & \overline{\omega} & 0 & \omega & 1 & \omega & \omega & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \overline{\omega} & \overline{\omega} & 0 & \omega & 1 & \omega & \omega & 0 & 1 & \overline{\omega} & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \omega & \overline{\omega} & \omega & \omega & 0 & \overline{\omega} & 1 & \overline{\omega} & \overline{\omega} & 0 & 1 & \omega & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \omega & \omega & 0 & \overline{\omega} & 1 & \overline{\omega} & \overline{\omega} & 0 & 1 & \omega & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \overline{\omega} & \omega & \omega & 0 & 1 & \overline{\omega} & \overline{\omega} & \omega & 1 & \omega & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & \overline{\omega} & 0 & 1 & \omega & \omega & 1 & 0 & \overline{\omega} & 1 & \omega & \overline{\omega} & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & \omega & 1 & \overline{\omega} & \omega & \omega & \omega & \overline{\omega} & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ \omega & 0 & \omega & 1 & 1 & 0 & \overline{\omega} & \omega & \omega & 1 & \overline{\omega} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ \omega & \omega & \overline{\omega} & 1 & \omega & 1 & 1 & 0 & 0 & \overline{\omega} & 0 & \omega & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \omega & 0 & \overline{\omega} & \overline{\omega} & \overline{\omega} & 0 & 0 & \overline{\omega} & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Table 8.1: New Codes $\mathscr{C}_{\text{P}}$ over $\mathbb{F}_7$

#	q^m	m	r_1	ℓ	$\Gamma(L, x^{r_1-1})$	Codes $\mathscr{C}_{\text{P}}$	Codes in (Grassl, 2007)
$\mathscr{C}_1$	49	2	9	3	$[46,33,9]_7$	$[53,33,13]_7$	$[53,33,12]_7$
$\mathscr{C}_2$	49	2	17	3	$[46,22,17]_7$	$[53,22,21]_7$	$[53,22,20]_7$
$\mathscr{C}_3$	49	2	9	4	$[45,32,9]_7$	$[54,32,14]_7$	$[54,32,13]_7$
$\mathscr{C}_4$	49	2	17	4	$[45,21,17]_7$	$[54,21,22]_7$	$[54,21,21]_7$
$\mathscr{C}_5$	49	2	1	5	$[44,44,1]_7$	$[55,44,7]_7$	$[55,44,6]_7$
$\mathscr{C}_6$	49	2	9	5	$[44,31,9]_7$	$[55,31,15]_7$	$[55,31,14]_7$
$\mathscr{C}_7$	49	2	1	9	$[40,40,1]_7$	$[59,40,11]_7$	$[59,40,10]_7$

where ω is a primitive element of $\mathbb{F}_4$. The parity check matrix of $\mathscr{C}_{\text{P}}$ in reduced echelon form is thus

$$H_{\text{P}} = \begin{bmatrix}
1&0&0&0&0&0&0&0&0&0&0&\overline{\omega}&0&1&\overline{\omega}&1&1&\omega&0&\omega&\overline{\omega}\\
0&1&0&0&0&0&0&0&0&0&0&\omega&1&1&0&\overline{\omega}&0&\omega&1&\omega&1\\
0&0&1&0&0&0&0&0&0&0&0&\omega&\omega&\overline{\omega}&1&0&\omega&0&0&\omega&\omega\\
0&0&0&1&0&0&0&0&0&0&0&0&\omega&0&\omega&\omega&\omega&\omega&\omega&0&1\\
0&0&0&0&1&0&0&0&0&0&0&\omega&\overline{\omega}&\omega&1&1&\omega&1&\overline{\omega}&0&\omega\\
0&0&0&0&0&1&0&0&0&0&0&0&1&\overline{\omega}&\overline{\omega}&\overline{\omega}&\omega&1&\overline{\omega}&\omega&\overline{\omega}\\
0&0&0&0&0&0&1&0&0&0&0&\overline{\omega}&0&0&\omega&1&\omega&\omega&1&\omega&\omega\\
0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&1&0&\omega&\omega\\
0&0&0&0&0&0&0&0&1&0&0&\overline{\omega}&0&1&\overline{\omega}&\overline{\omega}&\overline{\omega}&\omega&0&\omega&0\\
0&0&0&0&0&0&0&0&0&1&0&\omega&\omega&\overline{\omega}&1&\omega&\overline{\omega}&1&1&0&\omega\\
0&0&0&0&0&0&0&0&0&0&1&\overline{\omega}&0&0&\omega&\omega&\overline{\omega}&\omega&\overline{\omega}&\omega&\omega
\end{bmatrix}.$$

Since $\deg G(x) = 5$, $m = 2$, $\ell = 4$ and $|L| = 12$, the code has parameters $[21,10,7]_4$. The minimum weight of the code was confirmed by direct computation using Magma (Bosma et al., 1997). Observe that the code $\mathscr{C}_{\text{P}}$ is an extension of the shortened BCH code $[12,10,2]_4$ defined with Goppa polynomial x and the set L.

8.5 Nested Structure From Codes $\mathscr{C}_{\text{P}}$

Consider the code $\mathscr{C}_{\text{R}}$ defined with the Goppa polynomial

$$G(x) = x^{r_1} \prod_{i=0}^{\ell-2} (x - \alpha^i)$$

and the parity check matrix

Table 8.2: New Codes $\mathscr{C}_P$ over $\mathbb{F}_8$

#	q^m	m	r_1	ℓ	$\Gamma(L, x^{r_1-1})$	Codes $\mathscr{C}_{\text{P}}$	Codes in (Grassl, 2007)
$\mathscr{C}_8$	64	2	10	3	$[61,46,10]_8$	$[68,46,14]_8$	$[68,46,13]_8$
$\mathscr{C}_9$	64	2	19	3	$[61,33,19]_8$	$[68,33,23]_8$	$[68,33,22]_8$
$\mathscr{C}_{10}$	64	2	28	3	$[61,22,28]_8$	$[68,22,32]_8$	$[68,22,31]_8$
$\mathscr{C}_{11}$	64	2	10	4	$[60,45,10]_8$	$[69,45,15]_8$	$[69,45,14]_8$
$\mathscr{C}_{12}$	64	2	19	4	$[60,32,19]_8$	$[69,32,24]_8$	$[69,32,23]_8$
$\mathscr{C}_{13}$	64	2	10	5	$[59,44,10]_8$	$[70,44,16]_8$	$[70,44,15]_8$
$\mathscr{C}_{14}$	64	2	19	5	$[59,31,19]_8$	$[70,31,25]_8$	$[70,31,24]_8$
$\mathscr{C}_{15}$	64	2	10	6	$[58,43,10]_8$	$[71,43,17]_8$	$[71,43,16]_8$
$\mathscr{C}_{16}$	64	2	19	6	$[58,30,19]_8$	$[71,30,26]_8$	$[71,30,25]_8$
$\mathscr{C}_{17}$	64	2	10	7	$[57,42,10]_8$	$[72,42,18]_8$	$[72,42,17]_8$
$\mathscr{C}_{18}$	64	2	1	8	$[56,56,1]_8$	$[73,56,10]_8$	$[73,56,9]_8$
$\mathscr{C}_{19}$	64	2	1	10	$[54,54,1]_8$	$[75,54,12]_8$	$[75,54,11]_8$
$\mathscr{C}_{20}$	64	2	1	11	$[53,53,1]_8$	$[76,53,13]_8$	$[76,53,12]_8$
$\mathscr{C}_{21}$	64	2	1	12	$[52,52,1]_8$	$[77,52,14]_8$	$[77,52,13]_8$
$\mathscr{C}_{22}$	64	2	1	13	$[51,51,1]_8$	$[78,51,15]_8$	$[78,51,14]_8$

$$H_{\text{R}} = \begin{bmatrix} 11\ldots1 & 1 & 0 & 0 & \cdots & 0 \\ H_{r_1} & 0 & 0 & 0 & \cdots & 0 \\ H_0 & 0 & h_0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{\ell-2} & 0 & 0 & 0 & \cdots & h_{\ell-2} \end{bmatrix}. \tag{8.9}$$

The submatrices H_i, $0 \le i \le \ell-2$ are $1 \times q^m - \ell$ rows of the form

$$H_i = \begin{bmatrix} \frac{1}{\alpha^{\ell-1}-\alpha^i} & \frac{1}{\alpha^{\ell}-\alpha^i} & \frac{1}{\alpha^{\ell+1}-\alpha^i} & \cdots & \frac{1}{\alpha^{q^m-2}-\alpha^i} \end{bmatrix},$$

where α is a primitive element of the field $\mathbb{F}_{q^m}$ and the submatrices h_i, $0 \le i \le \ell-2$ are $1 \times m$ rows of the form

$$h_i = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{m-1} \end{bmatrix}.$$

Table 8.3: New Codes $\mathscr{C}_{\text{P}}$ over $\mathbb{F}_9$

#	q^m	m	r_1	ℓ	$\Gamma(L, x^{r_1-1})$	Codes $\mathscr{C}_{\text{P}}$	Codes in (Grassl, 2007)
$\mathscr{C}_{23}$	81	2	11	3	$[78,61,11]_9$	$[85,61,15]_9$	$[85,61,14]_9$
$\mathscr{C}_{24}$	81	2	21	3	$[78,46,21]_9$	$[85,46,25]_9$	$[85,46,24]_9$
$\mathscr{C}_{25}$	81	2	31	3	$[78,33,31]_9$	$[85,33,35]_9$	$[85,33,34]_9$
$\mathscr{C}_{26}$	81	2	41	3	$[78,22,41]_9$	$[85,22,45]_9$	$[85,22,44]_9$
$\mathscr{C}_{27}$	81	2	11	4	$[77,60,11]_9$	$[86,60,16]_9$	$[86,60,15]_9$
$\mathscr{C}_{28}$	81	2	21	4	$[77,45,21]_9$	$[86,45,26]_9$	$[86,45,25]_9$
$\mathscr{C}_{29}$	81	2	31	4	$[77,32,31]_9$	$[86,32,36]_9$	$[86,32,35]_9$
$\mathscr{C}_{30}$	81	2	1	5	$[76,76,1]_9$	$[87,76,7]_9$	$[87,76,6]_9$
$\mathscr{C}_{31}$	81	2	11	5	$[76,59,11]_9$	$[87,59,17]_9$	$[87,59,16]_9$
$\mathscr{C}_{32}$	81	2	21	5	$[76,44,21]_9$	$[87,44,27]_9$	$[87,44,26]_9$
$\mathscr{C}_{33}$	81	2	31	5	$[76,31,31]_9$	$[87,31,37]_9$	$[87,31,36]_9$
$\mathscr{C}_{34}$	81	2	1	6	$[75,75,1]_9$	$[88,75,8]_9$	$[88,75,7]_9$
$\mathscr{C}_{35}$	81	2	11	6	$[75,58,11]_9$	$[88,58,18]_9$	$[88,58,17]_9$
$\mathscr{C}_{36}$	81	2	21	6	$[75,43,21]_9$	$[88,43,28]_9$	$[88,43,27]_9$
$\mathscr{C}_{37}$	81	2	1	7	$[74,74,1]_9$	$[89,74,9]_9$	$[89,74,8]_9$
$\mathscr{C}_{38}$	81	2	11	7	$[74,57,11]_9$	$[89,57,19]_9$	$[89,57,18]_9$
$\mathscr{C}_{39}$	81	2	21	7	$[74,42,21]_9$	$[89,42,29]_9$	$[89,42,28]_9$
$\mathscr{C}_{40}$	81	2	1	8	$[73,73,1]_9$	$[90,73,10]_9$	$[90,73,9]_9$
$\mathscr{C}_{41}$	81	2	11	8	$[73,56,11]_9$	$[90,56,20]_9$	$[90,56,19]_9$
$\mathscr{C}_{42}$	81	2	1	9	$[72,72,1]_9$	$[91,72,11]_9$	$[91,72,10]_9$
$\mathscr{C}_{43}$	81	2	1	10	$[71,71,1]_9$	$[92,71,12]_9$	$[92,71,11]_9$
$\mathscr{C}_{44}$	81	2	1	11	$[70,70,1]_9$	$[93,70,13]_9$	$[93,70,12]_9$
$\mathscr{C}_{45}$	81	2	1	12	$[69,69,1]_9$	$[94,69,14]_9$	$[94,69,13]_9$
$\mathscr{C}_{46}$	81	2	1	13	$[68,68,1]_9$	$[95,68,15]_9$	$[95,68,14]_9$
$\mathscr{C}_{47}$	81	2	1	14	$[67,67,1]_9$	$[96,67,16]_9$	$[96,67,15]_9$
$\mathscr{C}_{48}$	81	2	1	15	$[66,66,1]_9$	$[97,66,17]_9$	$[97,66,16]_9$

Observe from Equations (8.5) and (8.9) that $\mathscr{C}_{\text{R}}$ is simply a shortened form of $\mathscr{C}_{\text{P}}$. Thus if $\mathscr{C}_{\text{R}}$ and $\mathscr{C}_{\text{P}}$ are defined with the same Goppa polynomial $G(x)$ then H_{R} corresponds to H_{P} with the submatrix H_{I_1} removed. The codes $\mathscr{C}_{\text{R}}$ have parameters,

$$\begin{aligned} &\text{length:} && n = |L| + m(\ell-1) + 1, \\ &\text{redundancy:} && n - k \le m\ell + r_{\text{BCH}} + 1, \\ &\text{distance:} && d \ge r_1 + \ell + 1 = r + 2. \end{aligned}$$

Suppose $\mathscr{C}_{\text{R1}}$ is defined with $G_1(x) = x^a \prod_{i=0}^{\ell-2}(x-\alpha^i)$ and $\mathscr{C}_{\text{R2}}$ is defined with $G_2(x) = x^b \prod_{i=0}^{\ell-2}(x-\alpha^i)$ with $a < b$, then

$$\mathscr{C}_{\text{R2}} \subset \mathscr{C}_{\text{R1}}$$

holds[3]. It is well known code that nested codes can be extended using Construction X. Recall Construction X from Theorem 2.1.

8.6 Theorem (Construction X (Sloane et al., 1972)). *If a linear code $\mathscr{C}_1$ with parameters $[n_1,k_1,d_1]_q$ has a subcode $\mathscr{C}_2$ with parameters $[n_2,k_2,d_2]_q$, then $\mathscr{C}_1$ is extendable to a code with parameters $[n_1+n, k_1, \min\{d_1+\delta, d_2\}]_q$ using an auxiliary code $[n, k_1-k_2, \delta]_q$.*

Table 8.4: New Codes From Construction X in $\mathbb{F}_7$

#	$\mathscr{C}_2$			$\mathscr{C}_1$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{49}$	8	5	$[53,31,14]_7$	10	5	$[53,27,16]_7$	$[5,4,2]_7$	$[58,31,16]_7$	$[58,31,15]_7$

8.6 Results

In this section we present results on codes obtained from the two construction methods.

8.6.1 New Codes From $\mathscr{C}_{\text{P}}$

We use Goppa polynomials with coefficients in $\mathbb{F}_{q^m}$ of the form

$$G(x) = x^{r_1} \prod_{i=0}^{\ell-2}(x-\alpha^i),$$

where α is a primitive element of $\mathbb{F}_{q^m}$. The Goppa polynomial has $\deg G(x) = r = r_1 + \ell - 1$. Hence from Theorem 8.4 the codes $\mathscr{C}_{\text{P}}$ have minimum distance $d \ge r_1 + \ell + 1$. The codes presented in Tables 8.1–8.3 have minimum distances better than

[3]Notice however that $\mathscr{C}_{\text{P2}} \nsubseteq \mathscr{C}_{\text{P1}}$ since $\mathscr{C}_{\text{P2}} H_{\text{P1}}^T \neq \mathbf{0}$ when $\mathscr{C}_{\text{P1}}$ and $\mathscr{C}_{\text{P2}}$ are defined by $G_1(x)$ and $G_2(x)$ respectively and T is the transpose operator.

Table 8.5: New Codes From Construction X in $\mathbb{F}_8$

#	$\mathscr{C}_{R1}$			$\mathscr{C}_{R2}$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{50}$	9	3	$[66,46,13]_8$	11	3	$[66,42,15]_8$	$[5,4,2]_8$	$[71,46,15]_8$	$[71,46,14]_8$
$\mathscr{C}_{51}$	9	3	$[66,46,13]_8$	12	3	$[66,40,16]_8$	$[8,6,3]_8$	$[74,46,16]_8$	$[74,46,15]_8$
$\mathscr{C}_{52}$	13	3	$[66,38,17]_8$	18	3	$[66,33,22]_8$	$[9,5,5]_8$	$[75,38,22]_8$	$[75,38,21]_8$
$\mathscr{C}_{53}$	18	3	$[66,33,22]_8$	20	3	$[66,29,24]_8$	$[5,4,2]_8$	$[71,33,24]_8$	$[71,33,23]_8$
$\mathscr{C}_{54}$	18	3	$[66,33,22]_8$	21	3	$[66,27,25]_8$	$[8,6,3]_8$	$[74,33,25]_8$	$[74,33,24]_8$
$\mathscr{C}_{55}$	9	4	$[67,45,14]_8$	11	4	$[67,41,16]_8$	$[5,4,2]_8$	$[72,45,16]_8$	$[72,45,15]_8$
$\mathscr{C}_{56}$	13	4	$[67,37,18]_8$	18	4	$[67,32,23]_8$	$[9,5,5]_8$	$[76,37,23]_8$	$[76,37,22]_8$
$\mathscr{C}_{57}$	18	4	$[67,32,23]_8$	20	4	$[67,28,25]_8$	$[5,4,2]_8$	$[72,32,25]_8$	$[72,32,24]_8$
$\mathscr{C}_{58}$	9	5	$[68,44,15]_8$	11	5	$[68,40,17]_8$	$[5,4,2]_8$	$[73,44,17]_8$	$[73,44,16]_8$

the codes with the same length and dimension in (Grassl, 2007). The codes are represented in the form $[n,k,d]_q$. The dimensions of the codes in Tables 8.1–8.3 are obtained by expressing their respective parity check matrices in reduced echelon form.

8.6.2 New Codes From $\mathscr{C}_R$

Using Construction X on the codes $\mathscr{C}_{R1}$ and $\mathscr{C}_{R2}$ as defined in Section 8.5 and short optimal auxiliary codes, we are able to obtain 30 improvements to the tables in (Grassl, 2007) for the fields $\mathbb{F}_7$, $\mathbb{F}_8$ and $\mathbb{F}_9$. The results are shown in Tables 8.4–8.6. In addition to the 79 codes presented in Tables 8.1–8.6 many codes that improve the tables in (Grassl, 2007) can be obtained by shortening and puncturing codes in Tables 8.1–8.6.

8.7 Further Extensions of the Codes $\mathscr{C}_P$

The codes $\mathscr{C}_P$ defined by the parity check matrix in Equation (8.5) can be denoted as $\mathscr{C}_P[L,G]$ since they are defined for a coordinate set L and a Goppa polynomial $G(x)$. In (Tomlinson et al., 2011) the authors show that it is possible to extend the coordinate set L and produced new binary codes as a result. The method is applied for the case of nonbinary codes. A new coordinate set $\tilde{L}$ is defined,

$$\tilde{L} = R \cup L \qquad R \subseteq \{\alpha^i : G(\alpha^i) = 0, \alpha^i \in \mathbb{F}_{2^m} \setminus \{0\}\}.$$

which contains a subset of the roots of $G(x)$ except 0. This has the effect of increasing both the length and dimension of the code by $|R|$. A modified parity check matrix with the set $\tilde{L}$ and $G(x)$ is formed,

$$H_{\mathrm{P}[\tilde{L},G]} = \begin{bmatrix} 11..1 & 1 & 0 & 0 & \cdots & 0 \\ H_{r_1} & 0 & H_{\mathrm{I}} & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{r_\ell} & 0 & 0 & 0 & \cdots & H_{\mathrm{I}} \end{bmatrix} \tag{8.10}$$

where the row matrices H_{I} are as previously defined. As a consequence of modifying the coordinate set the matrices H_{r_μ} for μ such that $\beta_\mu \in R$ will each have an entry $\frac{1}{\beta_\mu - \beta_\mu}$ which is replaced with a zero. Normally each coordinate corresponding to $\beta_u \in R$ is deleted from all parity check equations of H_P. Replacing $\frac{1}{\beta_u - \beta_u}$ with a zero deletes the coordinates only for the affected parity check equations H_u, $\beta_u \in R$ by multiplying these coordinates by zero. The codes $\mathscr{C}_{\mathrm{P}}$ are labelled as *intermediate* codes.

8.7 ***Theorem.*** *If the coordinate set of the code $\mathscr{C}_P$ is appended with a set R and undefined entries ($\frac{1}{\beta_\mu - \beta_\mu}$) of the intermediate parity check matrix are replaced with a zero, the intermediate binary code $\mathscr{C}_{P[\tilde{L},G]}$ has minimum distance $d \geq d_P - |R|$ where d_P is the minimum distance of $\mathscr{C}_P$. Furthermore all codewords of weight w such that $d_P - |R| \leq w < d_P$ are nonzero in at least one of the coordinates in R.*

Proof. From the proof of Theorem 8.4, it is clear that any v rows H_{r_μ} of the parity check matrix of $\mathscr{C}_{\mathrm{P}}$ contribute v to the distance of the code d_{P}. Consider the intermediate code, all codewords that are zero in the coordinates specified by R in the intermediate parity check are codewords of the original code $\mathscr{C}_{\mathrm{P}}$ and have distance d_{P}. All codewords that are nonzero in at least one coordinate in R are not in the original code $\mathscr{C}_{\mathrm{P}}$ thus for these codewords the rows of the intermediate parity check matrix that have an entry $\frac{1}{\beta_\mu - \beta_\mu}$ replaced with a zero are not guaranteed to contribute to their minimum weight. There are $|R|$ of such rows, hence the minimum weight is $\geq d_{\mathrm{P}} - |R|$. ∎

With the knowledge that all codewords of weight $< d_{\mathrm{P}}$ in the intermediate code $\mathscr{C}_{P[\tilde{L},G]}$ are nonzero in at least one of the coordinates specified by R, it is possible to extend the minimum distance of $\mathscr{C}_{P[\tilde{L},G]}$ by simply repeating each of these coordinates in R so that the minimum distance is increased to d_{P} and the length is

increased by $|R|$. The new code is called a *modified* code denoted by $\mathscr{C}_{P_m}$. These modified codes $\mathscr{C}_{P_m}$ have parameters,

$$\begin{aligned}
&\text{length:} && n \le |L| + m\ell + 1 + 2|R| = |\tilde{L}| + m\ell + 1 + |R| \\
&\text{redundancy:} && n - k \le m \deg G(x) + 1 + |R| \\
&\text{distance:} && d \ge \deg G(x) + 2
\end{aligned}$$

It is evident from Tables 8.7-8.8 that all the codes obtainable for this method are also obtainable from Construction X and codes with the same parameters can be found in Tables 8.4-8.5.

8.8 Summary

Construction P by Sugiyama et al. (1976) for binary Goppa codes was generalised to the case of nonbinary codes. The concept of an extended Goppa code was used to obtain improvements to the tables of best known codes in (Grassl, 2007). These codes can be seen as extended BCH codes and the method can be considered as an efficient construction of extended BCH codes. In total 48 new codes with better minimum distances than any known codes with the same length and dimension were obtained in finite fields $\mathbb{F}_7$, $\mathbb{F}_8$ and $\mathbb{F}_9$. In addition 30 new codes were found from further extensions using Construction X. Many more codes can be obtained from the new codes by simple modifications like shortening and puncturing codes in Tables 8.1-8.6.

Table 8.6: New Codes From Construction X in $\mathbb{F}_9$

#	$\mathscr{C}_{\text{R1}}$			$\mathscr{C}_{\text{R2}}$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{59}$	10	3	$[83,61,14]_9$	12	3	$[83,57,16]_9$	$[5,4,2]_9$	$[88,61,16]_9$	$[88,61,15]_9$
$\mathscr{C}_{60}$	10	3	$[83,61,14]_9$	13	3	$[83,55,17]_9$	$[8,6,3]_9$	$[91,61,17]_9$	$[91,61,16]_9$
$\mathscr{C}_{61}$	15	3	$[83,51,19]_9$	20	3	$[83,46,24]_9$	$[9,5,5]_9$	$[92,51,24]_9$	$[92,51,23]_9$
$\mathscr{C}_{62}$	20	3	$[83,46,24]_9$	22	3	$[83,42,26]_9$	$[5,4,2]_9$	$[88,46,26]_9$	$[88,46,25]_9$
$\mathscr{C}_{63}$	20	3	$[83,46,24]_9$	23	3	$[83,40,27]_9$	$[8,6,3]_9$	$[91,46,27]_9$	$[91,46,26]_9$
$\mathscr{C}_{64}$	24	3	$[83,38,28]_9$	30	3	$[83,33,34]_9$	$[10,5,6]_9$	$[93,38,34]_9$	$[93,38,33]_9$
$\mathscr{C}_{65}$	30	3	$[83,33,34]_9$	32	3	$[83,29,36]_9$	$[5,4,2]_9$	$[88,33,36]_9$	$[88,33,35]_9$
$\mathscr{C}_{66}$	10	4	$[84,60,15]_9$	12	4	$[84,56,17]_9$	$[5,4,2]_9$	$[89,60,17]_9$	$[89,60,16]_9$
$\mathscr{C}_{67}$	10	4	$[84,60,15]_9$	13	4	$[84,54,18]_9$	$[8,6,3]_9$	$[92,60,18]_9$	$[92,60,17]_9$
$\mathscr{C}_{68}$	11	4	$[84,58,16]_9$	14	4	$[84,52,19]_9$	$[8,6,3]_9$	$[92,58,19]_9$	$[92,58,18]_9$
$\mathscr{C}_{69}$	15	4	$[84,50,20]_9$	20	4	$[84,45,25]_9$	$[9,5,5]_9$	$[93,50,25]_9$	$[93,50,24]_9$
$\mathscr{C}_{70}$	19	4	$[84,46,24]_9$	23	4	$[84,39,28]_9$	$[10,7,4]_9$	$[94,46,28]_9$	$[94,46,27]_9$
$\mathscr{C}_{71}$	20	4	$[84,45,25]_9$	22	4	$[84,41,27]_9$	$[5,4,2]_9$	$[89,45,27]_9$	$[89,45,26]_9$
$\mathscr{C}_{72}$	20	4	$[84,45,25]_9$	23	4	$[84,39,28]_9$	$[8,6,3]_9$	$[92,45,28]_9$	$[92,45,27]_9$
$\mathscr{C}_{73}$	10	5	$[85,59,16]_9$	12	5	$[85,55,18]_9$	$[5,4,2]_9$	$[90,59,18]_9$	$[90,59,17]_9$
$\mathscr{C}_{74}$	10	5	$[85,59,16]_9$	13	5	$[85,53,19]_9$	$[8,6,3]_9$	$[93,59,19]_9$	$[93,59,18]_9$
$\mathscr{C}_{75}$	20	5	$[85,44,26]_9$	22	5	$[85,40,28]_9$	$[5,4,2]_9$	$[90,44,28]_9$	$[90,44,27]_9$
$\mathscr{C}_{76}$	10	6	$[86,58,17]_9$	12	6	$[86,54,19]_9$	$[5,4,2]_9$	$[91,58,19]_9$	$[91,58,18]_9$
$\mathscr{C}_{77}$	20	6	$[86,43,27]_9$	22	6	$[86,39,29]_9$	$[5,4,2]_9$	$[91,43,29]_9$	$[91,43,28]_9$
$\mathscr{C}_{78}$	10	7	$[87,57,18]_9$	12	7	$[87,53,20]_9$	$[5,4,2]_9$	$[92,57,20]_9$	$[92,57,19]_9$
$\mathscr{C}_{79}$	10	8	$[88,56,19]_9$	12	8	$[88,52,21]_9$	$[5,4,2]_9$	$[93,56,21]_9$	$[93,56,20]_9$

#	q^m	m	r_1	ℓ	Codes $\mathscr{C}_{\text{P}}$	$\lvert R\rvert$	Codes $\mathscr{C}_{P_m}$	Codes in (Grassl, 2007)
$\mathscr{C}_1$	49	2	9	6	$[56,30,16]_7$	1	$[58,31,16]_7$	$[58,31,15]_7$

Table 8.7: New Codes $\mathscr{C}_{P_m}$ in $\mathbb{F}_7$

#	q^m	m	r_1	ℓ	Codes $\mathscr{C}_{\text{P}}$	$\lvert R\rvert$	Codes $\mathscr{C}_{P_m}$	Codes in (Grassl, 2007)
$\mathscr{C}_2$	64	2	10	4	$[69,45,15]_8$	1	$[71,46,15]_8$	$[71,46,14]_8$
$\mathscr{C}_3$	64	2	11	4	$[69,43,16]_8$	1	$[71,44,16]_8$	$[71,44,15]_8$
$\mathscr{C}_4$	64	2	19	4	$[69,32,24]_8$	1	$[71,33,24]_8$	$[71,33,23]_8$
$\mathscr{C}_5$	64	2	20	4	$[69,30,25]_8$	1	$[71,31,25]_8$	$[71,31,24]_8$
$\mathscr{C}_6$	64	2	10	5	$[70,44,16]_8$	1	$[72,45,16]_8$	$[72,45,15]_8$
$\mathscr{C}_7$	64	2	10	5	$[70,44,16]_8$	2	$[74,46,16]_8$	$[74,46,15]_8$
$\mathscr{C}_8$	64	2	11	5	$[70,42,17]_8$	1	$[72,43,17]_8$	$[72,43,16]_8$
$\mathscr{C}_9$	64	2	19	5	$[70,31,25]_8$	1	$[72,32,25]_8$	$[72,32,24]_8$
$\mathscr{C}_{10}$	64	2	19	5	$[70,31,25]_8$	2	$[74,33,25]_8$	$[74,33,24]_8$
$\mathscr{C}_{11}$	64	2	10	6	$[71,43,17]_8$	1	$[73,44,17]_8$	$[73,44,16]_8$

Table 8.8: New Codes $\mathscr{C}_{P_m}$ in $\mathbb{F}_8$

#	q^m	m	r_1	ℓ	Ecodes $\mathscr{C}_{\text{P}}$	$\lvert R\rvert$	Ecodes $\mathscr{C}_{P_m}$	Ecodes in (Grassl, 2007)
$\mathscr{C}_{12}$	81	2	11	4	$[86,60,16]_9$	1	$[88,61,16]_9$	$[88,61,15]_9$
$\mathscr{C}_{13}$	81	2	12	4	$[86,58,17]_9$	1	$[88,59,17]_9$	$[88,59,16]_9$
$\mathscr{C}_{14}$	81	2	14	4	$[86,54,19]_9$	1	$[88,55,19]_9$	$[88,55,18]_9$
$\mathscr{C}_{15}$	81	2	21	4	$[86,45,26]_9$	1	$[88,46,26]_9$	$[88,46,25]_9$
$\mathscr{C}_{16}$	81	2	22	4	$[86,43,27]_9$	1	$[88,44,27]_9$	$[88,44,26]_9$
$\mathscr{C}_{17}$	81	2	31	4	$[86,32,36]_9$	1	$[88,33,36]_9$	$[88,33,35]_9$
$\mathscr{C}_{18}$	81	2	11	5	$[87,59,17]_9$	1	$[89,60,17]_9$	$[89,60,16]_9$
$\mathscr{C}_{19}$	81	2	11	5	$[87,59,17]_9$	2	$[91,61,17]_9$	$[91,61,16]_9$
$\mathscr{C}_{20}$	81	2	12	5	$[87,57,18]_9$	1	$[89,58,18]_9$	$[89,58,17]_9$
$\mathscr{C}_{21}$	81	2	12	5	$[87,57,18]_9$	2	$[91,59,18]_9$	$[91,59,17]_9$
$\mathscr{C}_{22}$	81	2	13	5	$[87,55,19]_9$	2	$[91,57,19]_9$	$[91,57,18]_9$
$\mathscr{C}_{23}$	81	2	21	5	$[87,44,27]_9$	1	$[89,45,27]_9$	$[89,45,26]_9$
$\mathscr{C}_{24}$	81	2	21	5	$[87,44,27]_9$	2	$[91,46,27]_9$	$[91,46,26]_9$
$\mathscr{C}_{25}$	81	2	22	5	$[87,42,28]_9$	1	$[89,43,28]_9$	$[89,43,27]_9$
$\mathscr{C}_{26}$	81	2	22	5	$[87,42,28]_9$	2	$[91,44,28]_9$	$[91,44,27]_9$
$\mathscr{C}_{27}$	81	2	11	6	$[88,58,18]_9$	1	$[90,59,18]_9$	$[90,59,17]_9$
$\mathscr{C}_{28}$	81	2	11	6	$[88,58,18]_9$	2	$[92,60,18]_9$	$[92,60,17]_9$
$\mathscr{C}_{29}$	81	2	12	6	$[88,56,19]_9$	1	$[90,57,19]_9$	$[90,57,18]_9$
$\mathscr{C}_{30}$	81	2	12	6	$[88,56,19]_9$	2	$[92,58,19]_9$	$[92,58,18]_9$
$\mathscr{C}_{31}$	81	2	21	6	$[88,43,28]_9$	1	$[90,44,28]_9$	$[90,44,27]_9$
$\mathscr{C}_{32}$	81	2	21	6	$[88,43,28]_9$	2	$[92,45,28]_9$	$[92,45,27]_9$
$\mathscr{C}_{33}$	81	2	21	6	$[88,43,28]_9$	3	$[94,46,28]_9$	$[94,46,27]_9$
$\mathscr{C}_{34}$	81	2	10	7	$[89,58,18]_9$	1	$[91,59,18]_9$	$[91,59,17]_9$
$\mathscr{C}_{35}$	81	2	11	7	$[89,57,19]_9$	1	$[91,58,19]_9$	$[91,58,18]_9$
$\mathscr{C}_{36}$	81	2	11	7	$[89,57,19]_9$	2	$[93,59,19]_9$	$[93,59,18]_9$
$\mathscr{C}_{37}$	81	2	12	7	$[89,55,20]_9$	1	$[91,56,20]_9$	$[91,56,19]_9$
$\mathscr{C}_{38}$	81	2	20	7	$[89,43,28]_9$	1	$[91,44,28]_9$	$[91,44,27]_9$
$\mathscr{C}_{39}$	81	2	21	7	$[89,42,29]_9$	1	$[91,43,29]_9$	$[91,43,28]_9$
$\mathscr{C}_{40}$	81	2	10	8	$[90,57,19]_9$	1	$[92,58,19]_9$	$[92,58,18]_9$
$\mathscr{C}_{41}$	81	2	11	8	$[90,56,20]_9$	1	$[92,57,20]_9$	$[92,57,19]_9$
$\mathscr{C}_{42}$	81	2	11	9	$[91,55,21]_9$	1	$[93,56,21]_9$	$[93,56,20]_9$

Table 8.9: New Codes $\mathscr{C}_{P_m}$ in $\mathbb{F}_8$

9. A Special Case of Shortening Linear Codes

9.1 Introduction

The minimum distance of a linear error correcting code is an important measure of performance of the code. Therefore it is desirable to obtain a linear code with the maximum possible minimum distance d, given a code length n and code dimension k. Linear codes in tables (Grassl, 2007) and (Schimd and Shurer, 2004) have the best known minimum distance of any known codes with the same length and dimension. Many of the codes in these tables are obtained from other good codes using well known methods of constructing new codes from existing ones. New codes can be obtained from existing ones by examining the low weight codewords of known optimal codes. Previous methods that have used this approach include Grassl and White (Grassl and White, 2004) in which a method of puncturing codes which uses the notion of a *hitting set* was presented and extending the length of codes by Kohnert (Kohnert, 2009) using a method that solves a set of Diophantine equations. In this Chapter a method of shortening linear codes in carefully chosen coordinates obtained by examining low weight codewords is presented. It is shown that this shortening method produces codes with parameters $[n-l, k-l, \geq d+\delta+1]$ when there are l deleted coordinates for some δ. It is also shown that these l coordinates can be used to extend the codes so that they form codes with parameters $[n+(\delta+1)l, k, d+\delta+1]$. The method is most efficient when the codes have a special structure. Recent results from Bezzateev and Shekhunova (Bezzateev and Shekhunova, 2008) on chains of binary Goppa codes provides codes with such a structure. Using the relationship between shortening and lengthening, four new binary codes with parameters $[243, 124, 33]_2$, $[244, 124, 34]_2$, $[245, 124, 35]_2$, and $[246, 124, 36]_2$ are presented.

9.2 Background

Shortened codes are obtained by deleting information symbols of a longer codes. A shortening method that has proved effective in producing good codes is constructions Y1 (MacWilliams and Sloane, 1983). Construction Y1 produces a shortened $[n-l, k-l+1, \geq d]_q$ code from a code $\mathscr{C}$ with parameters $[n, k, d]_q$ length n, dimension k and minimum distance d,defined in a finite field with cardinality q if the

dual of $\mathscr{C}$ has a codeword with weight l. In (Lim and Guan, 2006) a code shortening technique was presented for binary BCH codes and product codes. By discarding all the minimum weight codewords of a code with parameters $[n,k,d]_q$ the authors were able to produce a shortened code with parameters $[n-l,k-l,d+1]_q$. A code with parameters $[n,k,d]_q$ punctured in s coordinates has new parameters $[n-s,k,d-s]_q$. Grassl and White(Grassl and White, 2004) presented a puncturing scheme which in its simplest form punctures the code in coordinates that coincide with only zero symbols of any minimum weight codeword thus forming an $[n-s,k,d-s+1]_q$ code. By examining codewords of weight $\leq d+j$ the authors presented a generalised puncturing method that produced codes with parameters $[n-s,k,d-s+j]_q$ and as a result many new codes that have better distances than codes in (Grassl, 2007) with the same code rate were presented. Kohnert (Kohnert, 2009) presented a code extension scheme which extends a code $[n,k,d]_q$ to a code with parameters $[n+l,k,d+1]_q$. The appended l coordinates are simply repeated coordinates of the original code and coincide with at least one non-zero symbol in any minimum weight codewords. Kohnert (Kohnert, 2009) suggests that the presented lengthening scheme can be seen as an inverse of the puncturing scheme by Grassl (Grassl and White, 2004). In this Chapter a shortening scheme which can be seen as a generalisation of the method in (Lim and Guan, 2006) is presented. This method produces an $[n-l,k-l,\geq d+\delta+1]_q$ code from a $[n,k,d]_q$ code by examining codewords of weight up to $d+\delta$. In addition the set of l coordinates used in this method can be used to extend the original code to form an $[n+l,k,d+\delta+1]$ code. Using this relationship and some well studied Goppa codes from (Bezzateev and Shekhunova, 2008) improvements to the tables in (Grassl, 2007) are presented.

9.3 Code Shortening and Extension

Let $\mathscr{C}$ be a linear code of length n, dimension k and minimum distance d. Let the set $\{0,\ldots,n-1\}$ be the coordinates of the code $\mathscr{C}$. Shortening involves deleting $l<k$ information coordinates from the set $\{0,\ldots,n-1\}$. These $l<k$ information coordinates correspond to any linearly independent columns of the parity check matrix of $\mathscr{C}$. To ensure that the deleted coordinates are in fact information symbols Theorem 9.1 is employed.

9.1 Theorem (From (MacWilliams and Sloane, 1983)). *A code with minimum distance d has every combination of $d-1$ or less columns of its parity check matrix linearly independent.*

By constraining the number of deleted coordinates l such that $l<k$ and $l<d$ it is ensured that these l coordinates are information coordinates. Let $\mathbf{c}\in\mathscr{C}$ a codeword

of $\mathscr{C}$ then the support of $\mathbf{c}$ is defined as ,

$$supp(\mathbf{c}) = \{i : i \in \{0, \ldots, n-1\} \mid c_i \neq 0\}.$$

Let d be the minimum distance of the code and the weight of a codeword be the cardinality of its support,

$$weight(\mathbf{c}) = |supp(\mathbf{c})|.$$

Let $M \subset \mathscr{C}$ be the set of minimum weight codewords

$$M = \{\mathbf{c} \in \mathscr{C} : weight(c) = d\}$$

and W a set of sets satisfying,

$$W = \{supp(\mathbf{c}) : \forall \mathbf{c} \in M\}. \tag{9.1}$$

9.1 *Definition (From (Grassl and White, 2004)).* *A hitting set $\mathscr{J} \subseteq \{0, \ldots, n-1\}$ of the set W is any set such that every set $w \in W$ intersects $\mathscr{J}$. Formally,*

$$|\mathscr{J} \cap w| \geq 1 \qquad \forall w \in W$$

The hitting set $\mathscr{J}$ for the set W is a set containing coordinates such that every codeword of minimum weight in $\mathscr{C}$ is non-zero in at least one of the coordinates contained in $\mathscr{J}$.

9.3.1 Code Shortening

9.2 *Theorem.* *If $\mathscr{J}$ is a hitting set of the set W, shortening the code $\mathscr{C}$ in coordinates specified by $\mathscr{J}$ will produce a linear code with parameters $[n-l, k-l, \geq d+1]$ where $l = |\mathscr{J}|$, $l < k$ and $l < d$.*

Proof. Since every codeword of minimum weight in $\mathscr{C}$ is non-zero in at least one coordinate contained $\mathscr{J}$ and coordinates of $\mathscr{J}$ contain only information coordinates, all the minimum weight codewords of $\mathscr{C}$ are discarded as a result of shortening. Thus the shortened code $\mathscr{C}_s$ has minimum weight $\geq d+1$. ■

An observation from Theorem 9.2 is that all codewords (not necessarily of minimum weight) that are nonzero in at least one coordinate in $\mathscr{J}$ are also no longer in the shortened code.

$$\text{if } \mathbf{c} \in D \text{ then } \mathbf{c} \notin \mathscr{C}_s$$

This suggests that it is possible to extend the definition of the hitting set to include codewords of $\mathscr{C}$ with weight at most $d+\delta$ for some δ. The sets M_i are first defined as,

$$M_i = \{\mathbf{c} \in \mathscr{C} : weight(c) = d+i\}$$

and the sets $\tilde{M}$ and $\tilde{W}$ as,

$$\tilde{M} = \bigcup_{i=0}^{\delta} M_i$$
$$\tilde{W} = \{supp(\mathbf{c}) : \forall \mathbf{c} \in \tilde{M}\}.$$

Let $\mathscr{J}_\delta$ denote the hitting set of $\tilde{W}$.

$$|\mathscr{J}_\delta \cap w| \geq 1 \qquad \forall w \in \tilde{W}$$

9.1 *Corollary (Generalized Shortening).* *If $\mathscr{J}_\delta$ is a hitting set of the set $\tilde{W}$, shortening the code $\mathscr{C}$ in coordinates specified by $\mathscr{J}_\delta$ will produce a linear code with parameters $[n-l, k-l, \geq d+\delta+1]$ where $l = |\mathscr{J}_\delta|$, $l < k$ and $l < d$.*

9.3.2 Code Extension

A code $\mathscr{C}$ with parameters $[n,k,d]$ can be extended by appending l new coordinates so that the extended code $\mathscr{C}_e$ has parameters $[n+l, k, d+1]$, if every codeword of minimum weight d in $\mathscr{C}$ has a weight at least one in the appended l coordinates. It is possible to extend a code in this manner by examining all codewords of minimum weight. Let $\mathscr{J}$ define the hitting set of the set W as in Section 9.3 with $l = |\mathscr{J}|$. By repeating the l coordinates of $\mathscr{C}$ contained in $\mathscr{J}$ for every codeword of $\mathscr{C}$, it is possible to increase the minimum distance to $\geq d+1$.

9.3 *Theorem (From (Kohnert, 2009)).* *If $\mathscr{J}$ is a hitting set of the set W, extending the code $\mathscr{C}$ by repeating coordinates specified by $\mathscr{J}$ will produce a linear code with*

parameters $[n+l, k, d+1]$ *where* $l = |\mathscr{J}|$.

Proof. Since every codeword of minimum weight in $\mathscr{C}$ is nonzero in at least one coordinate in $\mathscr{J}$, repeating the l coordinates contained in $\mathscr{J}$ ensures that the weight of any of these codewords increases to at least $d+1$. ■

Suppose $\mathbf{G} = [\mathbf{g}_j]$, $j = [0, \ldots, n-1]$ is the generator matrix of $\mathscr{C}$ where $\mathbf{g}_j$ is a column of $\mathbf{G}$. A generator matrix of the extended code $\mathbf{G}_e$ with parameters $[n+l, k, d+1]$ can be formed as such,

$$\mathbf{G}_e = \mathbf{G}|\acute{\mathbf{G}} \text{ where } \acute{\mathbf{G}} = [\mathbf{g}_j], \ j \in \mathscr{J}$$

and | denotes matrix concatenation. This idea was presented in (Kohnert, 2009) in which the authors refer to this type of extension as an $(l, 1)$-extension. It is possible to generalise this extension by examining codewords of weight up do $d+\delta$ of the code $\mathscr{C}$. Let $\mathscr{J}_\delta$ and $\tilde{W}$ be defined as in Section 9.3.1.

9.4 ***Theorem.*** *If* $\mathscr{J}_\delta$ *is a hitting set of the set* $\tilde{W}$*, extending the code* $\mathscr{C}$ *by repeating coordinates specified by* $\mathscr{J}_\delta$ *a number of* p *times will produce a linear code with parameters* $[n+pl, k, d+p]$ *where* $l = |\mathscr{J}_\delta|$ *and* $1 \leq p \leq \delta + 1$.

Proof. Since all codewords of $\mathscr{C}$ of weight w, $d \leq w \leq d+\delta$ are nonzero in at least one coordinate in $\mathscr{J}_\delta$, repeating the l coordinates p times ensures that these codewords have weight at least $w+p$. The minimum of these weights is $d+p$, thus the extended code has minimum weight $d+p$. ■

The extended code $\mathscr{C}_e$ has generator matrix G_e as,

$$\mathbf{G}_e = \mathbf{G}|\acute{\mathbf{G}}_1|\cdots|\acute{\mathbf{G}}_p$$
$$\text{where } \acute{\mathbf{G}}_i = [\mathbf{g}_j], \ i = [1, \ldots, p] \text{ and } j \in \mathscr{J}_\delta$$

9.2 ***Corollary.*** *If a code* $\mathscr{C}$ *with parameters* $[n, k, d]$ *can be shortened to a code* $\mathscr{C}_s$ *with parameters* $[n-l, k-l, \geq d+\delta+1]$ *with a coordinate set* $\mathscr{J}_\delta$*, then* $\mathscr{C}$ *can also be extended to a code* $\mathscr{C}_e$ *with parameters* $[n+pl, k, d+p]$*,* $1 \leq p \leq \delta+1$ *using the same coordinate set.*

Equation (9.2) gives a summary of Corollary 9.2 given an $[n, k, d]$ code and a set $\mathscr{J}_\delta$.

$$[n-|\mathcal{J}_\delta|,k-|\mathcal{J}_\delta|,\geq d+\delta+1] \xleftarrow{Shortening} [n,k,d] \xrightarrow{Extending} [n+(\delta+1)|\mathcal{J}_\delta|,k,d+\delta+1] \tag{9.2}$$

TABLE I
Codes $\Gamma(L_1,G_1)$ and $\Gamma(L_2,G_1)$

Codes	$G(x)$	L	Length	Dimension	Distance
$\Gamma(L_1,G_1)$	$x^{t-1}+1$	$\{\mathbb{F}_{t^2}\setminus\mathbb{F}_t\}\cup\{0\}$	t^2-t+1	$t^2-t-2\ell\left(t-\frac{3}{2}\right)$	$2t-1$
$\Gamma(L_2,G_1)$	$x^{t-1}+1$	$\{\mathbb{F}_{t^2}\setminus\mathbb{F}_t\}$	t^2-t	$t^2-t-2\ell\left(t-\frac{3}{2}\right)-1$	$2t+4$

Example 9.1: Consider the Hermitian code $\mathscr{C}_{\text{H}}$ defined in $\mathbb{F}_{16}$ with parameters $[64,6,24]_{16}$. The code is defined with divisors D and G given as

$$G = mP_\infty = 11P_\infty$$
$$D = P_0+P_2+\cdots+P_{63}$$

The trace code of $\mathscr{C}_{\text{H}}$ in $\mathbb{F}_2$ is denoted by $\mathscr{C}_{\text{T}}$. This code has parameters $[64,13,24]_2$ and is optimal (see tables in Grassl, 2007). The $\mathscr{C}_{\text{T}}$ code has weight enumerator

$$x^{64}+368x^{40}y^{24}+2560x^{36}y^{28}+2334x^{32}y^{32}+2560x^{28}y^{36}+368x^{24}y^{40}+y^{64}.$$

Using a random search on the supports of the minimum weight codewords of $\mathscr{C}_{\text{T}}$ the set $\mathcal{J}_0=\{0,6,8,11,12,14,46,50,54\}$ with $|\mathcal{J}_0|=9$ was found. Note that $\mathcal{J}_0$ may not be unique even for a fixed ordering of the points P_i. Shortening $\mathscr{C}_{\text{T}}$ in these coordinates using Theorem 9.2 produces a code with parameters $[55,4,28]_2$ which is also optimal. It can be observed that the shortened code has minimum distance much greater than the lower bound which is 25 since the next available weight (from the weight enumerator) is 28. Extending the code using Theorem 9.3 and $\mathcal{J}_0$ we obtain a $[73,13,25]_2$ code.

Example 9.2: Consider the BCH $[63,18,21]_2$ code. A random search on the supports of the codewords of this code with weights 21 and 22 produces the set

$$\mathcal{J}_1=\{10,11,28,30,32,33,36,38,57,58,59\}$$

with $|\mathcal{J}_1|=11$. Shortening the BCH code in these coordinates produces a code with parameters $[57,7,23]_2$.

ℓ	$\mathbb{F}_{t^2}$	$\Gamma(L_1,G_1)$	$\Gamma(L_2,G_1)$
2	$\mathbb{F}_{16}$	$[13,2,7]_2$	$[12,1,12]_2$
3	$\mathbb{F}_{64}$	$[57,17,15]_2$	$[56,16,20]_2$
4	$\mathbb{F}_{256}$	$[241,124,31]_2$	$[240,123,36]_2$
5	$\mathbb{F}_{1024}$	$[993,687,63]_2$	$[992,686,68]_2$

Table 9.2: Codes $\Gamma(L_1,G_1)$ and $\Gamma(L_2,G_1)$ for $2 \le \ell \le 5$

9.4 Goppa Codes

Recall the definition and description of Goppa codes from Section 3.4. A Goppa code is called *separable* if its defining polynomial has distinct non-repeated roots. Separable Goppa codes with Goppa polynomials with coefficients in the finite field $\mathbb{F}_{t^2}$ and $t = 2^\ell$ are of particular interest. These codes are defined in the finite field $\mathbb{F}_2$, a subfield of $\mathbb{F}_{t^2}$. In (Bezzateev and Shekhunova, 1995) Bezzateev and Shekhunova present results on Goppa codes defined by $G_1(x) = x^{t-1} + 1$ and showed that the minimum distance of these codes is exactly $d = 2t-1$. The dimension of these codes was proven in (Véron, 2005) to be,

$$k = t^2 - t - 2\ell\left(t - \frac{3}{2}\right)$$

and the codes have length $n = t^2 - t + 1$. These codes have a location set $L_1 = \{\mathbb{F}_{t^2} \setminus \mathbb{F}_t\} \cup \{0\}$. These codes are denoted as $\Gamma(L_1,G_1)$. In a separate paper (Bezzateev and Shekhunova, 2008), Bezzateev and Shekhunova showed that shortened codes $\Gamma(L_2,G_1)$ obtained from $\Gamma(L_1,G_1)$ with a set $L_2 = \{\mathbb{F}_{t^2} \setminus \mathbb{F}_t\}$ have minimum distance $d = 2t+4$. These binary shortened codes have parameters $[n-1,k-1,2t+4]_2$. Table I gives the parameters of these two codes. Table 9.2 shows the parameters of these two codes in the range $2 \le \ell \le 5$. The codes $\Gamma(L_1,G_1)$ may be extended using Corollary 9.2. Using this approach,

$$\mathscr{C} = \Gamma(L_1,G_1) \quad \mathscr{C}_s = \Gamma(L_2,G_1) \quad \delta = 2t+4-(2t-1)-1 = 4$$

in addition,

$$\mathscr{J}_\delta = \mathscr{J}_4 = \{0\} \qquad l = 1$$

if it is assumed that the location set is ordered such that $L_1 = \{0,\alpha_1,\ldots,\alpha_{n-1}\}$ where $\alpha_0 = 0$. This means that shortening the codes $\Gamma(L_1,G_1)$ with minimum distance $2t-1$ in the coordinate $\mathscr{J}_4 = \{0\}$ will increase the distance of the shortened code to $2t+4$. From Section 9.3.1 and Corollary 9.1 it is clear that all codewords of $\Gamma(L_1,G_1)$ of weight w such that $2t-1 \le w \le 2t+3$ are nonzero in location $\{0\}$. Consequently the code $\Gamma(L_1,G_1)$ can be extended using Corollary 9.2 to obtain codes $\mathscr{C}_e$ with pa-

ℓ	$\mathbb{F}_{t^2}$	$\Gamma(L_1,G_1)$	p	$\mathscr{C}_e$	Codes in (Grassl, 2007)
2	$\mathbb{F}_{16}$	$[13,2,7]_2$	1	$[14,2,8]_2$	$[14,2,9]_2$
			2	$[15,2,9]_2$	$[15,2,10]_2$
			3	$[16,2,10]_2$	$[16,2,10]_2$
			4	$[17,2,11]_2$	$[17,2,11]_2$
			5	$[18,2,12]_2$	$[18,2,12]_2$
3	$\mathbb{F}_{64}$	$[57,17,15]_2$	1	$[58,17,16]_2$	$[58,17,18]_2$
			2	$[59,17,17]_2$	$[59,17,19]_2$
			3	$[60,17,18]_2$	$[60,17,20]_2$
			4	$[61,17,19]_2$	$[61,17,20]_2$
			5	$[62,17,20]_2$	$[62,17,21]_2$
4	$\mathbb{F}_{256}$	$[241,124,31]_2$	1	$[242,124,32]_2$	$[242,124,32]_2$
			2	$\mathbf{[243,124,33]_2}$	$[243,124,32]_2$
			3	$\mathbf{[244,124,34]_2}$	$[244,124,33]_2$
			4	$\mathbf{[245,124,35]_2}$	$[245,124,34]_2$
			5	$\mathbf{[246,124,36]_2}$	$[246,124,35]_2$
5	$\mathbb{F}_{1024}$	$[993,687,63]_2$	1	$[994,687,64]_2$	–
			2	$[995,687,65]_2$	–
			3	$[996,687,66]_2$	–
			4	$[997,687,67]_2$	–
			5	$[998,687,68]_2$	–

Table 9.3: Codes $\mathscr{C}_e$ for $2 \le \ell \le 5$, $1 \le p \le 5$

rameters,

$$[t^2-t+1+pl,\ t^2-t-2\ell\left(t-\frac{3}{2}\right),\ 2t-1+p]_2 \quad 1 \le p \le 5$$

Example 9.3 (An Example in $\mathbb{F}_{64}$): Let $\ell = 3$, the code $\Gamma(L_1,G_1)$ is defined with a Goppa polynomial with coefficients in $\mathbb{F}_{64}$. The Goppa code has parameters $[57,17,15]_2$. Examining all the codewords of this code of weight w such that $15 \le w \le 19$ using the computer algebra system MAGMA (Bosma et al., 1997), it is evident that all of these codewords are nonzero in the location $\{0\}$. Shortening the code at location $\{0\}$ produced a $[56,16,20]_2$ code. By repeating the coordinate $\{0\}$ of the code $[57,17,15]_2$ up to 5 times the minimum distances of the codes $[58,17,16]_2$, $[59,17,17]_2$, $[60,17,18]_2$, $[61,17,19]_2$ and $[61,17,20]_2$ are verified.

Table 9.3 shows the extended codes $\mathscr{C}_e$ obtainable from the codes $\Gamma(L_1,G_1)$ for $1 \le \ell \le 5$. The codes in bold font have better minimum distances than codes in the tables (Grassl, 2007) with the same length and dimension. To obtain good codes one needs to find the weight distribution and an optimal hitting set (one with the least possible size). Finding the minimum distance of a code in the simplest of finite fields $\mathbb{F}_2$ was shown to be an NP-hard problem in (Vardy, 1997). Thus computing

the weight distribution of a code, a related problem, can also be considered difficult. Furthermore, computing the hitting set was shown in (Garey and Johnson, 1979) to be an NP-complete problem. For example computing the weight distribution of the $[241,124,31]_2$ code will require enumerating $2^{124} = 2.1268 \times 10^{37}$ codewords. However there is no need to do this using Corollary 9.2 and results from (Bezzateev and Shekhunova, 1995) (Bezzateev and Shekhunova, 2008) in order to obtain improved codes.

9.5 Alternative Method

The code extension method in Section 9.3.2 may not be the most efficient code extension method since every $|\mathscr{J}_\delta|$ increase in length increases the minimum distance by 1. Using the ubiquitous construction X it is possible to obtain better extensions. Construction X (Sloane et al., 1972) (described in Section 2.1) uses a code and its subcode to extend the original code. It is enough to show that any shortened code is a subcode of the original code in order to obtain codes from construction X for this case.

9.5 ***Theorem.*** *If a code $\mathscr{C}$ can be shortened to a code $\mathscr{C}_s$ in coordinates $R \subset \{0,\ldots,n-1\}$, then the code $\mathscr{C}_p$ obtained by inserting a 0 (padding) in every deleted coordinate in R for every codeword in $\mathscr{C}_s$ is a subcode of $\mathscr{C}$.*

The proof of the theorem is straightforward since every codeword in $\mathscr{C}$ satisfies the parity check equations of the code $\mathscr{C}_p$. Suppose a code $\mathscr{C}$ with parameters $[n,k,d]$ is shortened to a code $\mathscr{C}_s$ using Corollary 9.1 to a code with parameters $[n-|\mathscr{J}_\delta|,k-|\mathscr{J}_\delta|,d_s]$ where $d_s \geq d+\delta+1$, then a padded code $\mathscr{C}_p$ can be formed by inserting zeros in every coordinate in $\mathscr{J}_\delta$ for every codeword in $\mathscr{C}_s$. As $\mathscr{C}_p \subset \mathscr{C}$ it is possible to extend $\mathscr{C}$ to a code with parameters $[n+\acute{n},k,d+\acute{d}]$ with construction X using an auxiliary code $[\acute{n},\mathscr{J}_\delta,\acute{d}]$ provided $\acute{d} \leq d_s - d$. For the case of the Goppa codes in Section 9.4, it is possible to pad the codewords of $\Gamma(L_2,G_1)$ with parameters $[n-|\mathscr{J}_\delta|,k-|\mathscr{J}_\delta|,2t+4]_2$ in the coordinate $\mathscr{J}_4 = \{0\}$ to form a code $\mathscr{C}_p$ with parameters $[n-|\mathscr{J}_\delta|+1,k-|\mathscr{J}_\delta|,2t+4]_2$. Consequently $\mathscr{C}_p \subset \Gamma(L_1,G_1)$ where $\Gamma(L_1,G_1)$ has parameters $[n,k,2t-1]_2$. Using Construction X with auxiliary repetition codes $[\acute{n},1,\acute{n}]_2$ up to $1 \leq \acute{n} \leq 5$ will produce the codes in Table 9.3 with dimension k. The two extension methods will produce the same results for the case when $|\mathscr{J}_\delta| = 1$ since the size is minimal. However as $|\mathscr{J}_\delta|$ increases, construction X will produce better extensions.

9.6 Summary

A method of shortening linear codes whilst improving the minimum distance by examining low weight codewords of the code is presented. The relationship between shortening and extending linear codes is examined. Using this relationship four new binary codes from a well studied Goppa code are obtained with parameters $[243, 124, 33]_2$, $[244, 124, 34]_2$, $[245, 124, 35]_2$ and $[246, 124, 36]_2$. Since shortened codes can be viewed as subcodes of the original code, the shortening method is then a method of obtaining good subcodes of a linear code.

Part IV

More On Algebraic Codes

10. On Extending BCH Codes

Wolf (1969) showed that Reed Solomon codes can be extended by adding at most two columns to their parity check matrices. An $[n,k,d]_q$ RS code can be extended to a singly extended RS code with parameters $[n+1,k+1,d]_q$ code or a doubly extended RS code with parameters $[n+2,k+2,d]_q$ code. The motivation behind this Chapter was to find out the extendability of the subfield subcodes of RS codes namely BCH codes. It turns out that the extendability of a BCH code depends on its subcodes. The necessary conditions under which a BCH code defined in a finite field $\mathbb{F}_q$ with an extension field $\mathbb{F}_{q^m}$ having a code length n, dimension k and minimum distance d is extendable to a code of length $n+p(m+1)$, dimension k and minimum distance $d+\delta$ for some $p>1$ and $\delta>0$ are presented.

10.1 The Method

First a criterion on extending the length of any linear code $\mathscr{C}$ by adding a single parity check on a set of co-ordinates of $\mathscr{C}$ so that the minimum distance of the code increases by 1 is established. A linear code of length n is an n-dimensional vector space and has a set of co-ordinates $\{0,1,\ldots n-1\}$. If the code $\mathscr{C}$ has minimum distance d and the set S is defined as,

$$S = \{\mathbf{c} : \mathbf{c} \in \mathscr{C} \text{ and } weight(\mathbf{c}) = d\}$$

where $supp$ and $weight$ denote the support and weight of a codeword of $\mathscr{C}$ respectively.

10.1 ***Theorem.*** *If there exists a set of co-ordinates $l \subseteq \{0,1,\ldots n-1\}$ of a linear code $\mathscr{C}$ with length n, dimension k and minimum distance d (and thus has parameters $[n,k,d]$), such that every codeword of minimum weight has weight exactly 1 in these l co-ordinates, then by adding a single parity check on the l co-ordinates the code $\mathscr{C}$ can be extended to an $[n+1,k,d+1]$ code.*

Proof. The proof is straight forward. Since for all the minimum weight codewords the l co-ordinates have only 1 co-ordinate non-zero, a parity check on only these l

co-ordinates will also be non-zero for every codeword of minimum weight. Consequently all codewords of previous weight d will have weight $d+1$. ■

10.2 BCH Codes

Recall the well known BCH bound (MacWilliams and Sloane, 1983) for cyclic codes without proof. Let $\mathbb{F}_q$ be a finite field and $\mathbb{F}_{q^m}$ as its extension field. Let α be the primitive element of the extension field $\mathbb{F}_{q^m}$

10.2 ***Theorem (BCH Bound from (MacWilliams and Sloane, 1983)).*** *A cyclic code $\mathscr{C}$ having defined with $d-1$ cyclically consecutive elements of a finite field $\mathbb{F}_{q^m}$ as roots of the form,*

$$\{\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+d-2}\} \quad \alpha \in \mathbb{F}_{q^m}$$

has minimum distance at least d.

The BCH bound in Theorem 10.2 describes a lower bound on the weight of any codewords $\mathbf{c} \in \mathscr{C}$. Given any codeword $\mathbf{c} \in \mathscr{C}$, the BCH bound can be used to obtain a lower bound on the weight of $\mathbf{c}$.

10.1 ***Corollary.*** *Any codeword $\mathbf{c} \in \mathscr{C}$ having the set of cyclically consecutive elements of finite field $\mathbb{F}_{q^m}$ as*

$$\{\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+w-2}\} \quad \alpha \in \mathbb{F}_{q^m}$$

as roots has weight at least w.

Consider the parity check matrix of a Reed Solomon code with redundancy r and length n defined in a field $\mathbb{F}_{q^m}$ with α as a primitive element,

$$H_{\text{RS}} = \begin{bmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+r} & \alpha^{2(b+r)} & \cdots & \alpha^{(n-1)(b+r)} \end{bmatrix} \tag{10.1}$$

The RS code has a set of cyclically consecutive roots $V = \{\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+r}\}$ with cardinality r. A subfield subcode of this RS code is a BCH code restricted to $\mathbb{F}_q$. In addition to the consecutive roots of the RS code, the BCH code will have additional roots that are co-members with the consecutive roots in conjugacy classes defined by the Frobenius automorphism. Recall the definition of a conjugacy class. A conjugacy

class of an element β of a finite field $\mathbb{F}_{q^m}$ is given as the set,

$$C(\beta) = \{\beta, \beta^q, \beta^{q^2}, \ldots, \beta^{q^{(e-1)}}\} \quad \beta \in \mathbb{F}_{q^m} \tag{10.2}$$

where e is the smallest positive integer such that $\beta^{q^e} = \beta$. The set of roots of a BCH code are given by

$$R = \bigcup_{\beta \in V} C(\beta), \tag{10.3}$$

the codes have redundancy $|R|$ and dimension $k = n - |R|$. Clearly $V \subset R$ and the minimum distance of BCH codes is at least $|V| + 1 = r + 1$. Often R contains one or more roots that are cyclically consecutive to the set of roots in V. Let $T \subset R$ denote these additional set of consecutive roots with $T \nsubseteq V$ then the minimum distance of the BCH code is

$$d \geq |V| + |T| + 1$$

from the BCH bound. The notations $\sim$ and $\nsim$ are used to denote whether elements of a set are cyclically consecutive. Thus

$$\sim A$$

means that *all* elements in a set A are cyclically consecutive and

$$\nsim A$$

means that not *all* elements in A are cyclically consecutive. Similarly these notations are used to denote whether the union of two sets has all its elements cyclically consecutive. Thus A is consecutive to B is expressed as,

$$A \sim B \text{ if and only if } C = A \cup B \text{ and } \sim C$$

Clearly if $A \sim B$ then $B \sim A$. Also

$$A \nsim B \text{ if and only if } C = A \cup B \text{ and } \nsim C$$

Let a *gap* root β of the sets A and B be defined as a root neither in the set A nor in the set B but one which when included in either set makes the elements of the two sets consecutive. Formally,

$$\begin{aligned} &\text{if } A \nsim B,\ \beta \notin A\ ,\ \beta \notin B \\ &\text{then } (A \cup \{\beta\}) \sim B \\ &\text{or } (B \cup \{\beta\}) \sim A \end{aligned}$$

From the definition of a conjugacy class in Equation (10.2), it is possible express the parity check matrix of the BCH code H_{BCH} as in Equation (10.4),

$$H_{\text{BCH}} = \begin{bmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{bq} & \alpha^{2bq} & \cdots & \alpha^{(n-1)bq} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b(q^{m-1})} & \alpha^{2b(q^{m-1})} & \cdots & \alpha^{(n-1)bq^{m-1}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+r} & \alpha^{2(b+r)} & \cdots & \alpha^{(n-1)(b+r)} \\ 1 & \alpha^{(b+r)q} & \alpha^{2(b+r)q} & \cdots & \alpha^{(n-1)(b+r)q} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{(b+r)(q^{m-1})} & \alpha^{2(b+r)(q^{m-1})} & \cdots & \alpha^{(n-1)(b+r)q^{m-1}} \end{bmatrix} \tag{10.4}$$

H_{BCH} is the parity matrix obtained by restricting H_{RS} to $\mathbb{F}_q$. Let

$$V = \{\alpha^{b}, \alpha^{b+1}, \ldots, \alpha^{b+r}\}$$

be the set of consecutive roots of the RS code C_{RS}, R the set defined in Equation (10.3) and $T \subset R$, $T \nsubseteq V$ be a set such that $V \sim T$. The parameters of $\mathscr{C}_{\text{BCH}}$ are denoted as $[n,k,d]_q$. For the sake of clarity we recall from Section 3.3 some important sets for $\mathscr{C}_{\text{BCH}}$ that will be continually referred to. The set V contains consecutive roots of the RS code in $\mathbb{F}_{q^m}$ that defines $\mathscr{C}_{\text{BCH}}$ in $\mathbb{F}_q$, the set T is obtained from elements in the conjugacy classes of all the elements in V and has all its elements consecutive to the elements in V, and finally a set $D = V \cup T$. The BCH code $\mathscr{C}_{\text{BCH}}$ has minimum distance $d \geq |D| + 1$.

10.3 Single Extension

It is possible to extend the BCH code in the same manner as singly extended RS codes (Wolf, 1969). Consider the parity check matrix of a singly extended BCH code with an additional row and column as in Equation (10.5). A codeword $\mathbf{c}$ of $\mathscr{C}_{\text{BCH}}$ is represented as a polynomial in a univariate polynomial ring ,

$$\mathbf{c}(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}$$

therefore a codeword $\acute{\mathbf{c}}$ of $\mathscr{C}_{\text{EBCH}}$ will be

$$\acute{\mathbf{c}}(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} + x^{n} c_n$$

$$H_{\text{EBCH}} = \begin{bmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} & 0 \\ 1 & \alpha^{bq} & \alpha^{2bq} & \cdots & \alpha^{(n-1)bq} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{b(q^{m-1})} & \alpha^{2b(q^{m-1})} & \cdots & \alpha^{(n-1)bq^{m-1}} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{b+r} & \alpha^{2(b+r)} & \cdots & \alpha^{(n-1)(b+r)} & 0 \\ 1 & \alpha^{(b+r)q} & \alpha^{2(b+r)q} & \cdots & \alpha^{(n-1)(b+r)q} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{(b+r)(q^{m-1})} & \alpha^{2(b+r)(q^{m-1})} & \cdots & \alpha^{(n-1)(b+r)q^{m-1}} & 0 \\ \mathbf{1} & \alpha^{\mathbf{b+r+1}} & \alpha^{\mathbf{2(b+r+1)}} & \cdots & \alpha^{\mathbf{(n-1)(b+r+1)}} & \mathbf{1} \end{bmatrix} \tag{10.5}$$

with

$$c_n = \sum_{i=0}^{n-1} c_i (\alpha^{b+r+1})^i \tag{10.6}$$

from the last row of the parity check matrix of H_{EBCH}. For the purpose of this construction it is required that

$$\alpha^{b+r+1} \notin D = V \cup T \tag{10.7}$$

or more precisely $\alpha^{b+r+1} \notin T$. This is an essential requirement for Theorem 10.3. For Theorem 10.4 the condition in 10.7 ensures that α^{b+r+1} is a gap root of the two sets D and P where,

$$P \subset R \text{ and } \sim P.$$

10.4 Construction

Consider a codeword of the single extended BCH code $\mathscr{C}_{\text{EBCH}}$,

$$\acute{\mathbf{c}}(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1} + x^n c_n$$

with c_n defined in (10.6). Our construction involves restricting co-ordinates of $\mathscr{C}_{\text{EBCH}}$ in the range $[0 \ldots n-1]$ (i.e those that form codewords of $\mathscr{C}_{\text{BCH}}$) to the subfield $\mathbb{F}_q$ while on the additional co-ordinate n no such restriction is applied. In this way the symbol defined by c_n is in $\mathbb{F}_{q^m}$ and is represented as an m-length vector in $\mathbb{F}_q$. This is possible since $\mathbb{F}_{q^m}$ can be expressed as an m-dimensional vector space in $\mathbb{F}_q$ using a suitable map (Lidl and Niederreiter, 1986). The map π_m is defined as such,

$$\pi_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^m$$
$$\pi_m(\alpha^j) = (a_1, a_2, \ldots, a_m)^{\mathrm{T}} \quad \alpha^j \in \mathbb{F}_{q^m} \ , \ a_i \in \mathbb{F}_q.$$

which maps elements of $\mathbf{F}_{q^m}$ Suppose $[\gamma_1, \gamma_2, \ldots, \gamma_m]$ forms a suitable basis of the vector space $\mathbb{F}_q^m$, then $\alpha^j = a_1\gamma_1 + a_2\gamma_2 + \cdots + a_m\gamma_m$. A common choice for the basis is the *normal* basis.

$$[\alpha, \alpha^q, \ldots, \alpha^{q^{m-1}}] \quad \beta \in \mathbb{F}_{q^m}$$

which exists for any subfield of $\mathbb{F}_{q^m}$ (Lidl and Niederreiter, 1986). We define another map σ_m, Let $p(x)$ be a primitive polynomial of $\mathbb{F}_{q^m}$ over $\mathbb{F}_q$. The companion matrix of a polynomial $f(x) = a_0 + a_1x + \cdots + a_{m-1}x^{m-1} + x^m$ is defined as in Subsection 2.3.3.

The map σ_m is given by,

$$\begin{aligned}
&\sigma_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^{m\times m} \\
&\psi_m(\alpha^j) = C^j, \qquad \alpha^j \in \mathbb{F}_{q^m} \setminus \{0\}.
\end{aligned}$$

where α is the primitive element of $\mathbb{F}_{q^m}$. This map is denoted by σ_m,

$$\sigma_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^{m\times m}.$$

In summary each symbol in $\mathbb{F}_{q^m}$ can be represented by a unique $m \times m$ matrix. The parity check matrix of $\mathscr{C}_{\text{EBCH}}$ can then be expressed as,

$$H_{\text{EBCH}} = \begin{bmatrix}
\pi_m(1) & \pi_m(\alpha^b) & \cdots & \pi_m(\alpha^{(n-1)b}) & \sigma_m(0) \\
\pi_m(1) & \pi_m(\alpha^{b+1}) & \cdots & \pi_m(\alpha^{(n-1)(b+1)}) & \sigma_m(0) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\pi_m(1) & \pi_m(\alpha^{b+r}) & \cdots & \pi_m(\alpha^{(n-1)(b+r)}) & \sigma_m(0) \\
\pi_m(1) & \pi_m(\alpha^{b+r+1}) & \cdots & \pi_m(\alpha^{(n-1)(b+r+1)}) & \sigma_m(1)
\end{bmatrix} \tag{10.8}$$

10.3 *Theorem (Single Extension).* *The minimum weight of the code $\mathscr{C}_{EBCH}$ is $d+1$, where d is the minimum weight of $\mathscr{C}_{BCH}$.*

Proof. Consider again the codeword of $\mathscr{C}_{\text{EBCH}}$

$$\begin{aligned}
\acute{\mathbf{c}}(x) &= c_0 + c_1x + \cdots + c_{n-1}x^{n-1} + x^n \sum_{i=0}^{n-1} c_i(\alpha^{b+r+1})^i \\
&= \mathbf{c}(x) - x^n \sum_{i=0}^{n-1} c_i(\alpha^{b+r+1})^i
\end{aligned}$$

with $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$. Clearly the symbol,

$$-c_n = \sum_{i=0}^{n-1} c_i(\alpha^{b+r+1})^i = \mathbf{c}(\alpha^{b+r+1})$$

is an evaluation of the codewords of $\mathscr{C}_{\text{BCH}}$ at the root α^{b+r+1}. Let $\mathbf{c} \in S$ i.e. $weight(\mathbf{c}) = d$ then,

$$-c_n = \mathbf{c}(\alpha^{b+r+1}) = \sum_{i=0}^{n-1} c_i(\alpha^{b+r+1})^i \neq 0$$

since from Equation 10.7 it is known that α^{b+r+1} is not a root of the code $\mathscr{C}_{\text{BCH}}$ i.e. $\alpha^{b+r+1} \notin D$. Suppose α^{b+r+1} were a root of $\mathscr{C}_{\text{BCH}}$, then from the BCH bound in Theorem 10.2 the code $\mathscr{C}_{\text{BCH}}$ will have minimum weight $d+1$ which is not the case. This means that all codewords of minimum weight d in $\mathscr{C}_{\text{BCH}}$ are extended to codewords of minimum weight $\geq d+1$ in $\mathscr{C}_{\text{EBCH}}$ (since the symbol on the co-ordinate n of a codeword of $\mathscr{C}_{\text{EBCH}}$ is mapped to an m length vector representing c_n). If we assume that $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ is not a codeword of minimum weight i.e. $weight(c) \geq d+1$ then,

$$\mathbf{c}(\alpha^{b+r+1}) = \sum_{i=0}^{n-1} c_i(\alpha^{b+r+1})^i = 0$$

for codewords $\mathbf{c}$ which have $D \cup \{\alpha^{b+r+1}\}$ as roots. From Corollary 10.1 these codewords have weight at least $d+1$ since $\{\alpha^{b+r+1}\} \sim D$ and $|D \cup \{\alpha^{b+r+1}\}| = d$. Therefore the minimum weight codewords of $\mathscr{C}_{\text{EBCH}}$ consist of;

(1) Codewords $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ that have weight d appended with an m-length vector representing c_n with weight exactly 1[1].

(2) Codewords $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ that have weight $d+1$, α^{b+r+1} as a root and appended with an m-length vector representing c_n having weight exactly zero.

The minimum weight of $\mathscr{C}_{\text{EBCH}}$ is thus $d+1$. ■

Example 10.1: Consider the code restricted to $\mathbb{F}_2$ from the RS code defined in $\mathbb{F}_{32}$ therefore $m = 5$. The RS code has a set of defining roots

$$V = \{1, \alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6\}.$$

[1] If c_n maps to an m-length vector of weight greater than 1 then the codeword $\acute{\mathbf{c}}$ will have weight $\geq d+2$.

The conjugacy classes $\mathbb{F}_{32}$ over $\mathbb{F}_2$ are ,

$$\{1\}$$
$$\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$$
$$\{\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17}\}$$
$$\{\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^{18}\}$$
$$\{\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{25}, \alpha^{19}\}$$
$$\{\alpha^{11}, \alpha^{22}, \alpha^{13}, \alpha^{26}, \alpha^{21}\}$$
$$\{\alpha^{15}, \alpha^{30}, \alpha^{29}, \alpha^{27}, \alpha^{23}\}$$

where α is the primitive element of $\mathbb{F}_{32}$. The set R for the BCH code is thus,

$$R = \{1, \alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}, \alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17}, \alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^{18}\}$$

with $|R| = 16$ and $T = \emptyset$. The set D is

$$D = V \cup T = V$$
$$\text{and } |D| = 7.$$

The BCH code has length $n = q^m - 1 = 31$, dimension $k = n - |R| = 15$ and minimum distance $d = |D| + 1 = 8$. The parity check matrix of $\mathscr{C}_{\text{EBCH}}$ is,

$$H_{\text{EBCH}} = \begin{bmatrix} \pi_5(1) & \pi_5(1) & \dots & \pi_5(1) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha) & \dots & \pi_5(\alpha^{30}) & \sigma_5(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_5(1) & \pi_5(\alpha^6) & \dots & \pi_5(\alpha^{25}) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(1) \end{bmatrix}$$

which maps to

$$H_{\text{EBCH}} = \left[\begin{array}{cccccccccccccccccccccccccccccccccccc}
1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0\\
1&0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&0&0&0\\
0&1&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&0&0&0\\
0&0&1&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&0&0&0\\
0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0\\
0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0\\
1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&0&0&0\\
0&0&0&1&0&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&1&0&0&0&0\\
0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&0&0\\
0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&0&0&0&0\\
0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&0\\
1&0&0&0&0&1&1&0&1&0&1&1&1&0&0&0&1&0&0&1&0&1&1&1&1&0&0&0&1&1&0&1&0&0\\
0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&0&0&0&0\\
0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&1&0&0&1&0&1&1&0&0&0\\
0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&0&1&0&0&0&0&0&0\\
1&1&1&0&1&0&0&0&1&0&0&1&0&1&0&1&1&0&0&0&0&1&1&1&0&0&1&1&0&1&0&0&0&0\\
0&0&0&1&0&0&1&0&1&1&0&0&1&1&1&1&1&0&0&0&1&1&0&1&1&1&0&1&1&0&1&0&0&0\\
0&1&0&1&1&0&0&0&0&1&1&1&0&1&0&1&1&1&1&1&1&0&1&0&0&0&1&0&0&1&0&0&0&0\\
0&0&1&1&1&0&0&0&1&0&1&1&1&0&1&0&1&0&0&0&0&1&1&0&1&1&1&1&1&0&0&0&0&0\\
1&0&0&1&0&0&1&1&1&1&0&1&1&1&0&0&0&0&1&0&1&0&1&1&0&0&0&0&1&0&0&0&0&0\\
0&1&1&1&1&1&0&1&1&1&0&0&0&1&0&1&0&1&1&0&1&0&0&0&0&1&1&0&0&1&0&0&0&0\\
0&0&1&0&1&0&1&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&1&0&1&1&1&0&0&0&0\\
0&1&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&1&1&1&1&0&1&1&1&0&0&0&1&0&1&0&0&0\\
0&0&0&1&1&0&0&1&0&0&1&1&1&1&1&0&1&1&1&0&0&0&1&0&1&1&0&1&0&1&0&0&0&0\\
1&0&1&0&0&0&1&1&0&1&1&1&1&1&0&0&1&0&0&1&1&0&0&0&0&1&0&1&1&0&1&0&0&0\\
0&0&0&1&0&1&1&0&1&0&1&0&0&0&1&1&1&0&1&1&1&1&1&0&0&1&0&0&1&1&0&1&0&0\\
0&1&1&0&1&0&1&0&0&0&1&1&1&0&1&1&1&1&1&1&0&0&1&0&0&1&1&0&0&0&1&0&0&1&0\\
0&0&1&1&0&0&0&0&1&0&1&1&0&1&0&1&0&0&0&1&1&1&0&1&1&1&1&0&0&1&0&0&0&1&0\\
0&1&1&1&1&1&0&0&1&0&0&1&1&0&0&0&0&1&0&1&1&0&1&0&1&0&0&0&0&1&1&1&0&0&0&1
\end{array}\right]$$

and reduces to

$$H_{\text{EBCH}} = \left[\begin{array}{cccccccccccccccccccccccccccccccccccc}
1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&1&0&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&1&1&1&1&1&1\\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&0&1&1\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&1&1&0&0&1&0&1&1&1&1\\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&0&1&1&0&1&0&1&1&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&1&0&1&1&1&1&0&1&1\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&1&1&0&1&0&1&0&1&0\\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&1&1&1&0&1&1&1&0&1&1\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&1&1&1&1&1&1&0&1&1&0\\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&1&1&0&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1&1&1&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&1&0&1&0&1&0&1&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&1&1&0&1&1&1&0&1&1&1&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1&0&1&0&0&1&1&1&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&1&0&1&0&0&1&1&1&0&0&0&0&1&1&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&1&0&0&1&1&1&0&0&0&1&0&0&1&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&1&0&1&1&1&0&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0&0&1&0&0&1&0&1&1&0&1&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&1&0&0&0&0&1&0&1&0&1&0&1&0&1&0&0\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&0&0&0&0&1&1&0&0&1&0&0&1&0&1&1
\end{array}\right]$$

in echelon form. The code has length $n+m=36$ and dimension $k=15$. The minimum distance of this code was verified using computer algebra system *MAGMA* (Bosma et al., 1997) to be 9. This result agrees with Theorem 10.3. The code $\mathscr{C}_{\text{EBCH}}$ has parameters $[36,15,9]_2$ and is an extension of the code $\mathscr{C}_{\text{BCH}}$ with parameters $[31,15,8]_2$.

The code $\mathscr{C}_{\text{EBCH}}$ is therefore an $[n+m,k,d+1]_q$ code. The necessary conditions for which an $[n+m+1,k,d+2]_q$ code can be obtained from $\mathscr{C}_{\text{EBCH}}$ by adding a parity

check on the appended m length vector representing c_n are now looked at. Ideally a situation where c_n is non-zero[2] for all the minimum weight codewords of $\mathscr{C}_{\text{EBCH}}$ is desirable so that Theorem 10.1 can be applied.

10.4 ***Theorem (Single Extension with Parity Check).*** *The extended BCH code $\mathscr{C}_{EBCH}$ with parameters $[n+m, k, d+1]_q$ can be further extended to an $[n+m+1, k, d+2]_q$ code by adding a parity check on the appended m-length vector representing c_n of every codeword of $\mathscr{C}_{EBCH}$ if the original BCH code $\mathscr{C}_{BCH}$ has α^{b+r+2} as a root.*

Proof. For the sake of clarity recall from Theorem 10.3 the minimum weight codewords of $\mathscr{C}_{\text{EBCH}}$,

(1) Codewords $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ that have weight d with an appended m-length vector representing c_n with weight exactly 1.

(2) Codewords $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ that have weight $d+1$, α^{b+r+1} as a root and appended with an m-length vector representing c_n having weight exactly zero.

If $\mathscr{C}_{\text{BCH}}$ also contains α^{b+r+2} as a root i.e. all codewords have this root, then the set of minimum weight codewords corresponding to (2) above that have α^{b+r+1} as a root will have α^{b+r+2} as a root also. From Corollary 10.1, these codewords will have weight equal to $d+2$ since $D \sim \{\alpha^{b+r+1}, \alpha^{b+r+2}\}$ and $|D \cup \{\alpha^{b+r+1}, \alpha^{b+r+2}\}| = d+1$. Since $\mathscr{C}_{\text{EBCH}}$ has minimum weight $d+1$, all codewords of minimum weight in $\mathscr{C}_{\text{EBCH}}$ will be from the set (1) above. These codewords have the appended m-length vector representing c_n having weight exactly 1. From Theorem 10.1, it is evident that $\mathscr{C}_{\text{EBCH}}$ can then be extended to an $[n+m+1, k, d+2]_q$ code by adding a single parity check on the m-length vector. ■

Example 10.2: Consider again the BCH code described in Example 10.1. We see that the gap root is $\alpha^{b+r+1} = \alpha^7$ and by examining the set R it can be observed that $\alpha^8 \in R$, thus $P = \{\alpha^8\}$. From Theorem 10.4 it should be possible to extend the code $\mathscr{C}_{\text{EBCH}}$ from a $[36, 15, 9]_2$ code to a $[37, 15, 10]_2$ code by adding a single parity check on the last $m = 5$ coordinates since $D \sim \{\alpha^7, \alpha^8\}$. The parity check matrix H is that of the extended code $\mathscr{C}_{\text{EBCH}}$ with a single parity check on the m-length vector representing the last symbol,

[2]The corresponding m-length vector will have weight exactly 1 from Theorem 10.3.

$$H = \left[\begin{array}{c}
1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0 \\
0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,0 \\
0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0 \\
0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,0\,1\,1\,0\,0\,1\,0\,1\,1\,1\,1\,0 \\
0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,0\,1\,1\,0\,1\,0\,1\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,1\,1\,1\,1\,0\,1\,1\,1\,0\,1\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,1\,1\,1\,1\,1\,1\,0\,1\,1\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1\,1\,1\,0\,0\,1\,0\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,1\,1\,0\,1\,0\,1\,0\,0\,1\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,0\,1\,0\,0\,0\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,1\,0\,1\,0\,0\,0\,0\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,1\,0\,0\,0\,0\,1\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,1\,0\,0\,0\,1\,1\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,1\,1\,1\,1\,0\,0\,1\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,1\,0\,1\,1\,1\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,0\,0\,1\,1\,0\,0\,0\,1\,0\,0\,1\,0\,1\,1\,0\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,1\,1\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,1\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,0\,1\,0 \\
0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1
\end{array}\right]$$

the resulting code was verified to be a $[37, 15, 10]_2$ code which coincides with the best known code from the tables in (Grassl, 2007).

10.5 Theorem (Multiple Extensions). *If the code $\mathscr{C}_{BCH}$ contains elements of the set*

$$P = \{\alpha^{b+r+2}, \alpha^{b+r+3}, \ldots, \alpha^{b+r+\delta+1}\}\ ,\ P \subset R\ ,\ |P| = \delta \text{ and } \sim P$$

as roots then each codeword of $\mathbf{c}$ of $\mathscr{C}_{BCH}$ can be extended to

$$\acute{\mathbf{c}}(x) = \mathbf{c}(x) + x^n c_n + x^{n+1} c_{n+1} + \cdots + x^{n+t-1} c_{n+t-1}$$

where

$$c_n = c_{n+1} = \cdots = c_{n+t-1} = -\sum_{i=0}^{n-1} c_i (\alpha^{b+r+1})^i,$$

$1 \le t \le \delta + 1$ and $\acute{\mathbf{c}} \in \mathscr{C}_{EBCH}$ with the extended code $\mathscr{C}$ having length $n + mt$, dimension k and minimum distance at least $d + t$.

Proof. It is assumed as before that each c_j, $j \ge n$ is restricted to $\mathbb{F}_{q^m}$ and maps to an m-length vector in $\mathbb{F}_q$. Henceforth each c_j, $j \ge n$ is referred as c_n since they are all equal. Every codeword $\mathbf{c} \in \mathscr{C}_{\text{BCH}}$ has the elements of P as roots, thus any codeword of $\mathscr{C}_{\text{BCH}}$ having α^{b+r+1} as a root will have the set $D \cup (\{\alpha^{b+r+1}\} \cup P)$ as a set of consecutive roots. From Corollary 10.1 this codeword will have weight at least $d + \delta + 1$ since $|D \cup (\{\alpha^{b+r+1}\} \cup P)| = d + \delta$, therefore all codewords of $\mathscr{C}_{\text{BCH}}$ of weight w such that

$d \le w \le d+\delta$ cannot have c_n equal to zero. Thus for $1 \le t \le \delta$ each c_n is an m-length vector having weight exactly 1 and $\mathscr{C}_{\text{EBCH}}$ has minimum weight codewords of weight $d+t$ consisting of minimum weight codewords of $\mathscr{C}_{\text{BCH}}$ of weight d appended with t m-length vectors each representing c_n with weight exactly 1. Thus $\acute{\mathbf{c}}$ has weight $d+t$. If $t = \delta+1$, the minimum weight codewords of $\mathscr{C}_{\text{EBCH}}$ consists of codewords of $\mathscr{C}_{\text{BCH}}$ with weight $d+\delta+1$ having roots $D \cup (\{\alpha^{b+r+1}\} \cup P)$ and appended with $\delta+1$ m-length vectors representing c_n each all zero, and minimum weight codewords of $\mathscr{C}_{\text{BCH}}$ appended with $\delta+1$ m-length vectors representing c_n with weight exactly 1. Thus $\acute{\mathbf{c}}$ has weight $d+\delta+1$. For $t > \delta+1$ no improvement on the minimum distance is possible since codewords with roots $D \cup (\{\alpha^{b+r+1}\} \cup P)$ cannot be extended because they have α^{b+r+1} as a root. Therefore given any t in the range $1 \le t \le \delta+1$ the code $\mathscr{C}_{\text{EBCH}}$ has minimum weight at least $d+t$. ■

Example 10.3: Using the BCH code in Example 10.1, it can be observed that the set R contains the set $P = \{\alpha^8, \alpha^9, \alpha^{10}\}$ and D $(\{\alpha^7\} \cup P)$. Thus $\delta = |P| = 3$. Theorem 10.5 says it is possible to obtain codes $[36,15,9]_2$ if $t = 1$, $[41,15,10]_2$ if $t = 2$, $[46,15,11]_2$ if $t = 3$ and $[51,15,12]_2$ if $t = 4$. Since the $t = 1$ coincides with Example 10.1, codes for $t \ge 2$ are constructed. If $t = 2$, the parity check matrix of $\mathscr{C}_{\text{EBCH}}$ is given by,

$$H_{\text{EBCH}} = \begin{bmatrix} \pi_5(1) & \pi_5(1) & \dots & \pi_5(1) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha) & \dots & \pi_5(\alpha^{30}) & \sigma_5(0) & \sigma_5(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \pi_5(1) & \pi_5(\alpha^6) & \dots & \pi_5(\alpha^{25}) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(1) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(1) \end{bmatrix}$$

this code was verified to be a $[41,15,10]_2$ code. If $t = 3$, H_{EBCH} is

$$H_{\text{EBCH}} = \begin{bmatrix} \pi_5(1) & \pi_5(1) & \dots & \pi_5(1) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha) & \dots & \pi_5(\alpha^{30}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \pi_5(1) & \pi_5(\alpha^6) & \dots & \pi_5(\alpha^{25}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(1) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(1) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(1) \end{bmatrix}$$

this is a $[46, 15, 11]_2$ code. Finally if $t = 4$,

$$H_{\text{EBCH}} = \begin{bmatrix} \pi_5(1) & \pi_5(1) & \dots & \pi_5(1) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha) & \dots & \pi_5(\alpha^{30}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_5(1) & \pi_5(\alpha^6) & \dots & \pi_5(\alpha^{25}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(1) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(1) & \sigma_5(0) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(1) & \sigma_5(0) \\ \pi_5(1) & \pi_5(\alpha^7) & \dots & \pi_5(\alpha^{24}) & \sigma_5(0) & \sigma_5(0) & \sigma_5(0) & \sigma_5(1) \end{bmatrix}$$

which results in a $[51, 15, 12]_2$ code.

In Theorem 10.4, α^{b+r+1} is a gap root of the sets D and $P = \{\alpha^{b+r+2}\}$. Cases where α^{b+r+1} is a gap root to sets D and P with $|P| \geq 1$ are now treated.

10.6 Theorem (Multiple Extensions with Parity Checks). *If the code $\mathscr{C}_{BCH}$ contains elements of the set*

$$P = \{\alpha^{b+r+2}, \alpha^{b+r+3}, \dots, \alpha^{b+r+\delta+1}\}\ ,\ P \subset R\ ,\ |P| = \delta \text{ and } \sim P$$

as roots then each codeword of $\mathbf{c}$ of $\mathscr{C}_{BCH}$ can be extended to

$$\acute{\mathbf{c}}(x) = \mathbf{c}(x) + x^n c_n + x^{n+1} c_{n+1} + \cdots + x^{n+t-1} c_{n+t-1}$$

where

$$c_n = c_{n+1} = \cdots = c_{n+t-1} = -\sum_{i=0}^{n-1} c_i (\alpha^{b+r+1})^i,$$

$1 \leq t \leq \delta$ and $\acute{\mathbf{c}} \in \mathscr{C}_{EBCH}$ with the extended code $\mathscr{C}$ having length $n + mt + l$, dimension k and minimum distance at least $min\{d + t + l, d + \delta + 1\}$ by adding l single parity checks on any of the m-length vectors representing $c_j, j \geq n$.

Proof. From the proof of Theorem 10.5, it can be observed that for $1 \leq t \leq \delta$ all minimum weight codewords of $\mathscr{C}_{\text{EBCH}}$ have m-length vectors representing c_j, $j \geq n$ each with weight exactly 1. Thus by Theorem 10.1 each c_j, $j \geq n$ is a candidate for extension by adding a single parity check on any l m-length vectors. It can also be observed that no extensions are possible for codewords with weight $d + \delta + 1$ with roots $D \cup (\{\alpha^{b+r+1}\} \cup P)$, therefore the maximum possible minimum distance of any extension of the code $\mathscr{C}_{\text{BCH}}$ with Theorem 10.6 is $d + \delta + 1$. Thus the minimum distance of the code is $min\{d + \delta + 1, d + t + l\}$. ■

Example 10.4: Using the sets R,D and P of BCH code in Example 10.3, $t = 3$ and $l = 3$ are chosen. For $t = 3$ from Example 10.3 a $[46, 15, 9]_2$ code is obtained which is then extended by adding $l = 3$ single parity checks to each c_j, $j \geq n$. H is the parity check matrix of the extended code obtained from $\mathscr{C}_{\text{BCH}}$ using Theorem 10.6

$$H = \left[\begin{array}{c}
1000000000000000000000011010000010000000000001000000 \\
0100000000000000000000010111000010000000000000000001 \\
0010000000000000000000010001100010000000000000011000 \\
0001000000000000000000010010110010000000000001111000 \\
0000100000000000000000010011010010000000000001101000 \\
0000010000000000000000010011101110000000000001000001 \\
0000001000000000000000010011111010000000000001011001 \\
0000000100000000000000010011111010000000000000100001 \\
0000000010000000000000011111011111000000000001001001 \\
0000000001000000000000010101011100000000000001001000 \\
0000000000100000000000010101011100000000000001010000 \\
0000000000010000000000011111010100000000000001010000 \\
0000000000001000000000011111010100000000000001000000 \\
0000000000000100000000011101110110000000000000000001 \\
0000000000000010000000101001110000000000000011000 \\
0000000000000001000000101001110000000000000111000 \\
0000000000000000100000101001110000000000001101001 \\
0000000000000000010001100010010000000000000011001 \\
0000000000000000001000110001001000000000000001100000 \\
0000000000000000000101110000101000000000000010100000 \\
0000000000000000000011010000110000000000000101000 \\
0000000000000000000000000000000000000000000001111001 \\
0000000000000000000000000000000000000000000010000000 \\
0000000000000000000000000000000000000000000001000000 \\
0000000000000000000000000000000000000000000000100000 \\
0000000000000000000000000000000000000000000000010000 \\
0000000000000000000000000000000000000000000000001000 \\
0000000000000000000000000000000000000000000010000001 \\
0000000000000000000000000000000000000000000001000010 \\
0000000000000000000000000000000000000000000000100001 \\
0000000000000000000000000000000000000000000000010000 \\
0000000000000000000000000000000000000000000000001000 \\
0000000000000000000000000000000000000000000000000101 \\
0000000000000000000000000000000000000000000000000011
\end{array}\right]$$

in echelon form. The code has parameters $[49, 15, 12]_2$.

Example 10.5: Consider the code restricted to $\mathbb{F}_4$ from the RS code defined in $\mathbb{F}_{64}$ with $m = 3$. The RS code has a set of defining roots

$$V = \{\alpha^k : k \in [41 \,..\, 62]\}$$

The conjugacy classes $\mathbb{F}_{64}$ over $\mathbb{F}_4$ are ,

$$\{1\},$$
$$\{\alpha, \alpha^4, \alpha^{16}\},$$
$$\{\alpha^2, \alpha^8, \alpha^{32}\},$$
$$\{\alpha^3, \alpha^{12}, \alpha^{48}\},$$
$$\{\alpha^5, \alpha^{20}, \alpha^{17}\},$$
$$\{\alpha^6, \alpha^{24}, \alpha^{33}\},$$
$$\{\alpha^7, \alpha^{28}, \alpha^{49}\},$$
$$\{\alpha^9, \alpha^{36}, \alpha^{18}\},$$
$$\{\alpha^{10}, \alpha^{40}, \alpha^{34}\},$$
$$\{\alpha^{11}, \alpha^{44}, \alpha^{50}\},$$
$$\{\alpha^{13}, \alpha^{52}, \alpha^{19}\},$$
$$\{\alpha^{14}, \alpha^{56}, \alpha^{35}\},$$
$$\{\alpha^{15}, \alpha^{60}, \alpha^{51}\},$$
$$\{\alpha^{21}\},$$
$$\{\alpha^{22}, \alpha^{25}, \alpha^{37}\},$$
$$\{\alpha^{23}, \alpha^{29}, \alpha^{53}\},$$
$$\{\alpha^{26}, \alpha^{41}, \alpha^{38}\},$$
$$\{\alpha^{27}, \alpha^{45}, \alpha^{54}\},$$
$$\{\alpha^{30}, \alpha^{57}, \alpha^{39}\},$$
$$\{\alpha^{31}, \alpha^{61}, \alpha^{55}\},$$
$$\{\alpha^{42}\},$$
$$\{\alpha^{43}, \alpha^{46}, \alpha^{58}\}$$

where α is the primitive element of $\mathbb{F}_{64}$. The set R for the BCH code is thus,

$$\begin{aligned} R \ = \{&\alpha^3, \alpha^7, \alpha^{11}, \alpha^{12}, \alpha^{13}, \alpha^{14}, \alpha^{15}, \alpha^{19}, \alpha^{23}, \alpha^{26}, \alpha^{27}, \alpha^{28}, \alpha^{29}, \\ &\alpha^{30}, \alpha^{31}, \alpha^{35}, \alpha^{38}, \alpha^{39}, \alpha^{41}, \alpha^{42}, \alpha^{43}, \alpha^{44}, \alpha^{45}, \alpha^{46}, \alpha^{47}, \alpha^{48}, \\ &\alpha^{49}, \alpha^{50}, \alpha^{51}, \alpha^{52}, \alpha^{53}, \alpha^{54}, \alpha^{55}, \alpha^{56}, \alpha^{57}, \alpha^{58}, \alpha^{59}, \alpha^{60}, \alpha^{61}, \alpha^{62}\} \end{aligned}$$

with $|R| = 40$ and $T = \emptyset$. The set D is

$$D = V \cup T = V$$
$$\text{and } |D| = 22.$$

The BCH code has length $n = q^m - 1 = 63$, dimension $k = n - |R| = 23$ and minimum distance $d = |D| + 1 = 23$. By studying R it is clear that α^{40} is a gap root of the sets

D and $P = \{\alpha^{38}, \alpha^{39}\}$. The parity check matrix of $\mathscr{C}_{\text{EBCH}}$ is,

$$H_{\text{EBCH}} = \begin{bmatrix} \pi_3(1) & \pi_3(\alpha^{62}) & \dots & \pi_3(\alpha) & \sigma_3(0) & \dots & \sigma_3(0) \\ \pi_3(1) & \pi_3(\alpha^{61}) & \dots & \pi_3(\alpha^2) & \sigma_3(0) & \dots & \sigma_3(0) \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ \pi_3(1) & \pi_3(\alpha^{41}) & \dots & \pi_3(\alpha^{22}) & \sigma_3(0) & \dots & \sigma_3(0) \\ \pi_3(1) & \pi_3(\alpha^{40}) & \dots & \pi_3(\alpha^{23}) & \sigma_3(1) & \dots & \sigma_3(0) \end{bmatrix}$$

which is a $[66, 23, 24]_4$ code. Applying Theorem 10.6 a $[67, 23, 25]_4$ code is obtained which corresponds to the best known code from the tables in (Grassl, 2007).

10.5 Observations

If it is assumed that an overall parity check is included for each m extended symbols, this method can be seen as extending a BCH code with parameters $[n, k, d]_q$,

$$[n, k, d]_q \rightarrow [n + t(m+1), k, min(d + \delta + 1, d + 2t)]$$

where $\delta = |P|$ and $t \geq 1$. Clearly the minimum distance of the extended code is upper bounded by the $d + \delta + 1$ which is the minimum distance of the subcode of the original $[n, k, d]_q$ code with parameters $[n, k - m, d + \delta + 1]_q$. A well known method that extends a code based on the parameters of its subcode is Construction X (Sloane et al., 1972) (see also Theorem 2.1). Construction X extends a code $[n, k, d]_q$ with a subcode $[n, k-m, d+\delta+1]_q$ to form a code with parameters $[n+\acute{n}, k, min(d+\delta+1, d+s)]_q$ using an auxiliary code $[\acute{n}, m, s]_q$. A best known code is usually chosen as the auxiliary code. Codes obtainable from the two constructions are now compared. For an extended code $[n + t(m+1), k, min(d + \delta + 1, d + 2t)]$ obtained from extending BCH codes, it is assumed that δ is arbitrarily large. The best known auxiliary code for construction X with length $t(m+1)$ and dimension m is obtained from the database in MAGMA (Bosma et al., 1997). Thus by comparing $2t$ and s the possible increase in distance from the two different methods for extended codes of the same length can be seen. Figures 10.1-10.4 show the extensions possible from the previously presented method of extending BCH codes and construction X for codes in different fields. The plots show the effect of m and q on the two extension methods. When m and q are small the two methods are similar as the length of the extended code increases. When m is small and q is large, construction X produces codes with much better parameters. This is because the auxiliary codes used in construction X get better as the field size increases. When q is small and m is large construction X produces much better codes as the length of the extended code increases. When q is large and m is small, the two methods seem to produce codes with similar distances

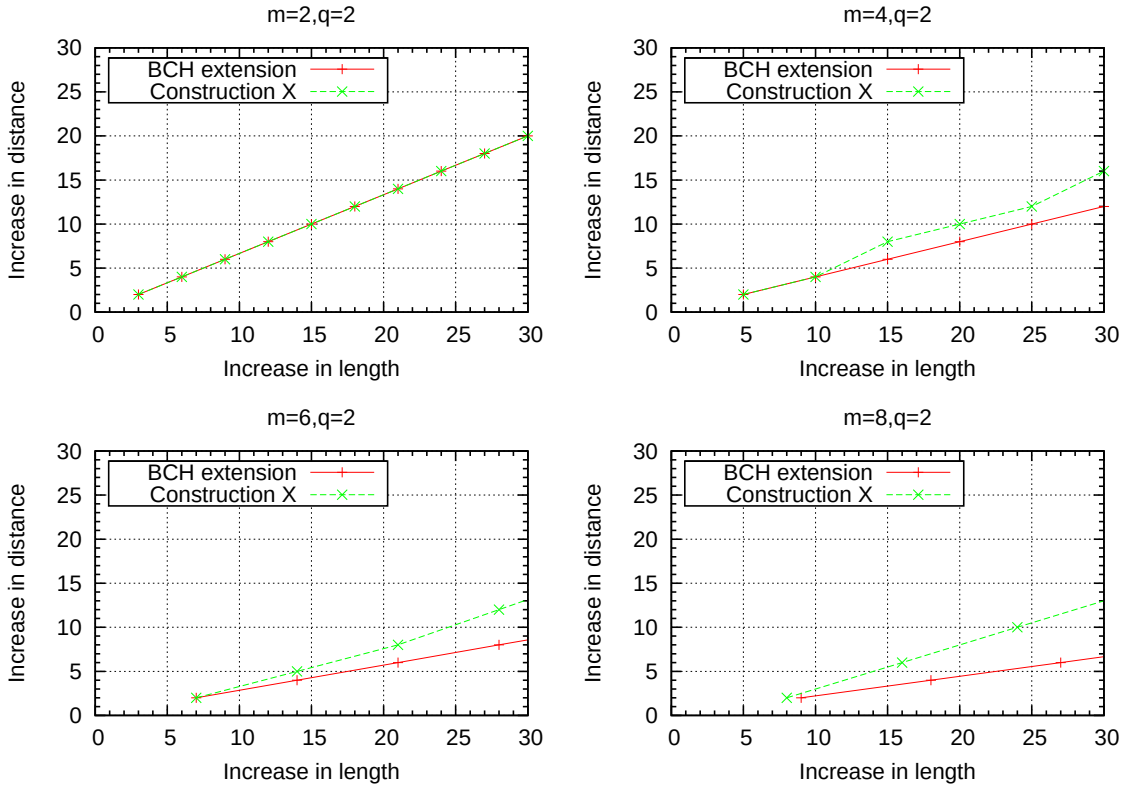

Fig. 10.1: BCH extension compared with Construction X in $\mathbb{F}_2$ for different m

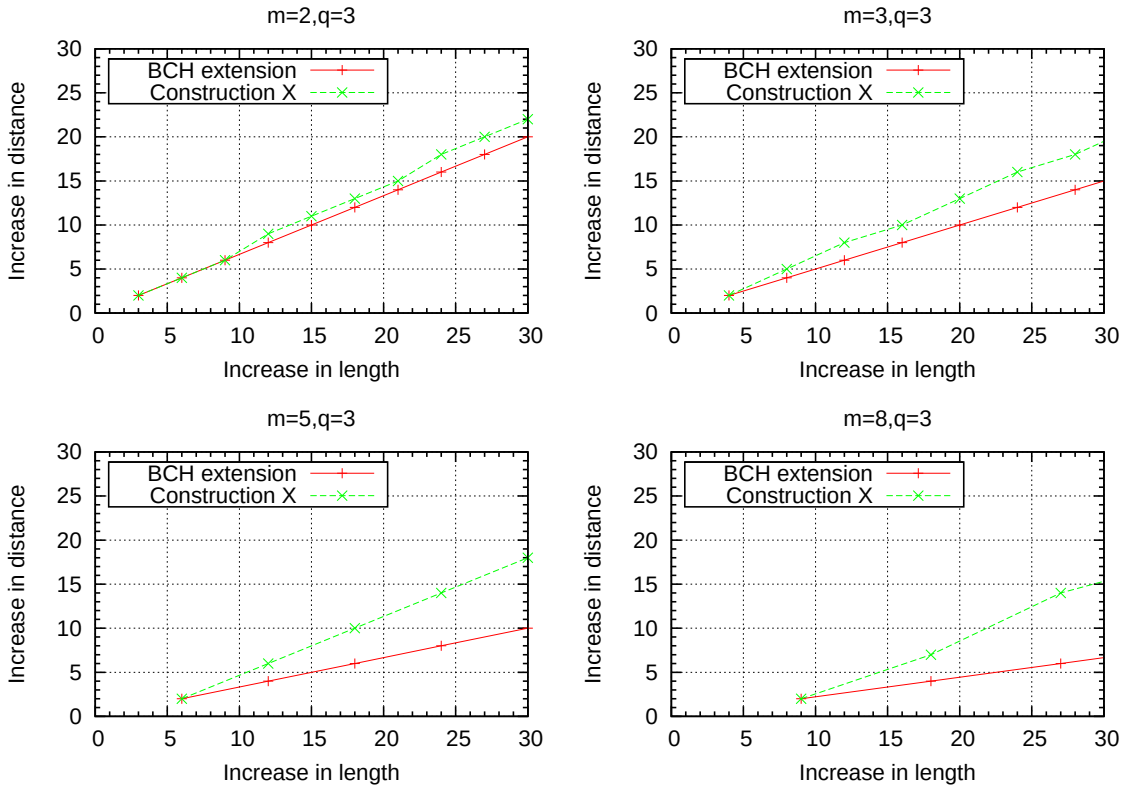

Fig. 10.2: BCH extension compared with Construction X in $\mathbb{F}_3$ for different m

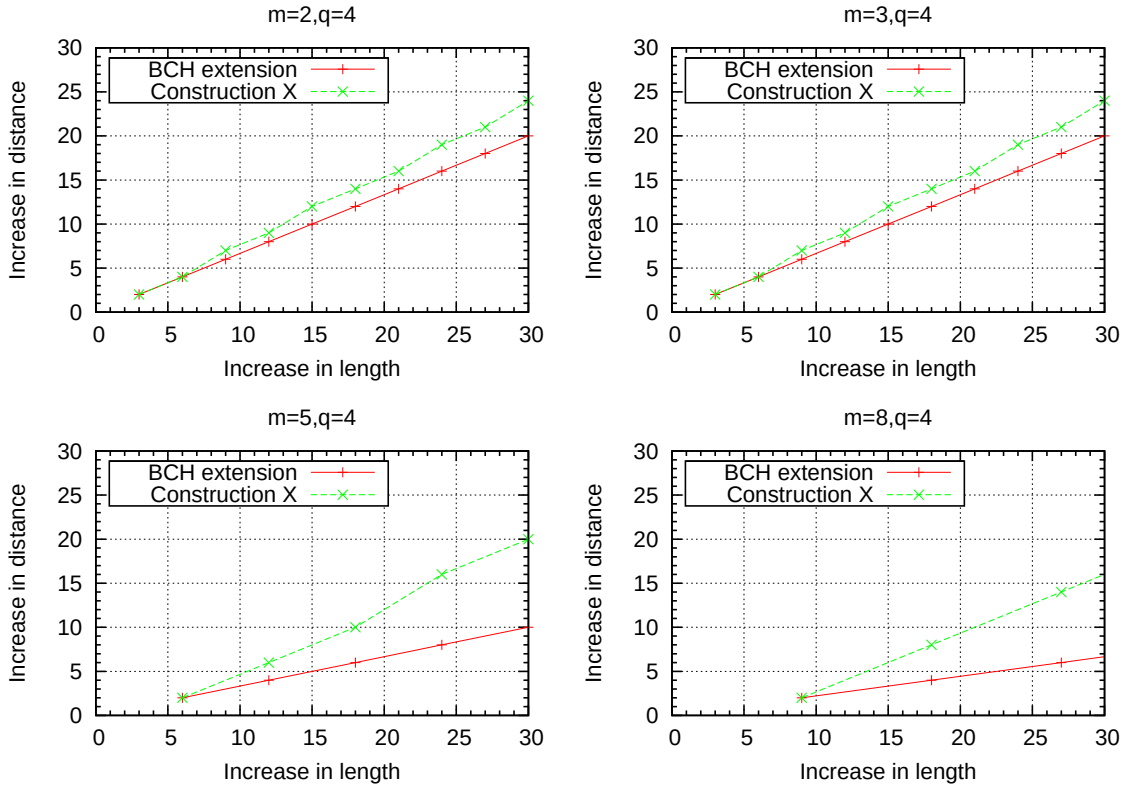

Fig. 10.3: BCH extension compared with Construction X in $\mathbb{F}_4$ for different m

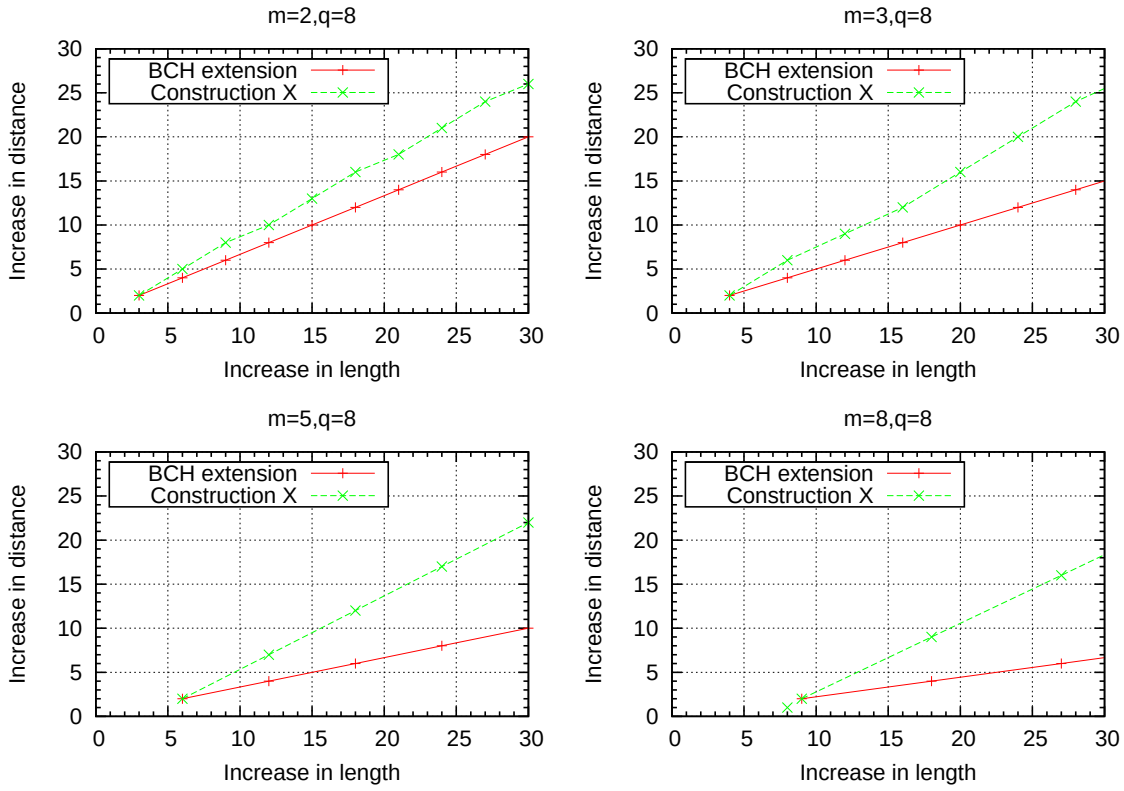

Fig. 10.4: BCH extension compared with Construction X in $\mathbb{F}_8$ for different m

when the difference between the length of the extended code and the original code is small. In the figures the difference in distance between the original code and the extended codes is allowed to go up to 30. In practice this difference is quite small as was seen in previous examples and the extended codes have lengths not much longer than the original codes. In summary when m is small this method of extending BCH codes produces similar results with construction X provided q is relatively small as well.

10.6 Summary

It was shown that it is possible to extend BCH codes by adding in some cases more than two columns to their parity check matrices whereas with RS codes it is only possible to add at most two columns. The extension method was shown to be as good as construction X in producing codes in some cases however as the field size increases the efficacy of the method deteriorates in comparison to construction X. The method has provided insight into the extendability of BCH codes. It has also provided insight into the inner workings of the method of extending Goppa codes in Chapter 8.

11. Improved Codes From Goppa Codes II

11.1 Introduction

In this chapter an alternative construction to the codes to the extended Goppa codes in Chapter 8 is given. Recall that the extended Goppa codes in Chapter 8 can be seen as extended BCH codes. The main advantage of the construction presented here to that in Chapter 8 is that it is possible extend non narrow-sense BCH codes. A drawback however is that for the codes in Chapter 8 with parameters $[n,k,d]_q$ the codes using the construction in this chapter will produce $[n-1,k-1,d]_q$ codes. In other words codes presented in this Chapter are shortened when compared with codes from Chapter 8. This is because the codes in Chapter 8 use the concept of a modified Goppa code. It is not known at present if this concept can be applied to the construction in this Chapter. The method presented here can be used for any BCH code (not just narrow sense BCH codes) thus providing better flexibility. The codes in the strict sense are not Goppa codes but alternant codes (a super-class of Goppa codes).

11.2 Goppa Codes

Recall the definition and description of Goppa codes from Section 3.4 and Section 8.2. In this chapter we assume the coordinate set L is indexed such that $L = \{\alpha_1, \ldots, \alpha_n\}$ and codewords of the Goppa code are indexed $c = (c_1, \ldots, c_n)$. The code defined in $\mathbb{F}_{q^m}$ which contains all the codewords of a Goppa code is a generalised Reed Solomon code and is maximum distance separable (MacWilliams and Sloane, 1983). The parity check matrix of a Goppa code, which by definition is restricted to $\mathbb{F}_q$, can be expressed with elements from $\mathbb{F}_q$. It is possible to represent $\mathbb{F}_{q^m}$ as an m-dimensional vector space with elements from $\mathbb{F}_q$ using a suitable basis. Let π_m define the map,

$$\pi_m : \mathbb{F}_{q^m} \rightarrow \mathbb{F}_q^m$$
$$\pi_m(\beta) = [a_1, a_2, \ldots, a_m] \quad \beta \in \mathbb{F}_{q^m} \ , \ a_i \in \mathbb{F}_q.$$

Suppose $[\gamma_1, \gamma_2, \ldots, \gamma_m]$ forms a suitable basis of the vector space $\mathbb{F}_q^m$, then $\beta = a_1\gamma_1 + a_2\gamma_2 + \cdots + a_m\gamma_m$. A common choice for the basis is the *normal* basis,

$$[\beta, \beta^q, \ldots, \beta^{q^{m-1}}] \quad \beta \in \mathbb{F}_{q^m}$$

which exists for any subfield of $\mathbb{F}_{q^m}$ (Lidl and Niederreiter, 1986). Given a parity check matrix defined in $\mathbb{F}_{q^m}$,

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,n} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{r,1} & h_{r,2} & \ldots & h_{r,n} \end{bmatrix}$$

it is possible to replace each $h_{i,j}$ in H with an m-tuple column vector $[h_{i,j,1}, h_{i,j,2}, \ldots, h_{i,j,m}]^T$. Finally performing row reductions on the $mr \times n$ matrix obtains a parity check matrix in $\mathbb{F}_q$ (MacWilliams and Sloane, 1983). The new matrix is a parity check matrix of the Goppa code in $\mathbb{F}_q$.

11.3 Construction

11.3.1 Preliminary: Cauchy and Vandermonde Matrices

Reed Solomon codes are maximum distance separable (MDS) codes and have the maximum achievable minimum distance. Their generator and parity check matrices are defined by a Vandermonde matrix. The parity check matrix of an RS code of length $n = |\mathbb{F}_q - 1|$, dimension k and minimum distance d defined in a finite field $\mathbb{F}_q$ with α as a primitive element is given by,

$$H_{\text{RS}} = \begin{bmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{b+n-k-1} & \alpha^{2(b+n-k-1)} & \cdots & \alpha^{(n-1)(b+n-k-1)} \end{bmatrix} \tag{11.1}$$

for some integer b. The parity check matrix of a doubly extended Reed Solomon (RS) code was shown by Wolf (1969) and Kasami *et al* (1966) to be

$$H_{\text{ERS}} = \begin{bmatrix} 1 & \alpha^b & \alpha^{2b} & \cdots & \alpha^{(n-1)b} & 1 & 0 \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & \alpha^{b+n-k-1} & \alpha^{2(b+n-k-1)} & \cdots & \alpha^{(n-1)(b+n-k-1)} & 0 & 1 \end{bmatrix} \tag{11.2}$$

Doubly extended RS codes have length $n+2$, dimension $k+2$ and distance d. Wolf (Wolf, 1969) proved the minimum distance of the codes defined by H_{ERS} have distance d by showing that each $d-1\times d-1$ determinant of the matrix is non-zero. This follows from the definition of minimum distance; all $d-1\times d-1$ submatrices of a parity check matrix of a code with minimum distance d must have a nonzero determinant. The determinant of any $d-1\times d-1$ submatrix of the Vandermonde matrix H_{RS} (Wolf, 1969) is given by,

$$\det\begin{vmatrix} \alpha^{bj_1} & \alpha^{bj_2} & \cdots & \alpha^{bj_{d-1}} \\ \alpha^{(b+1)j_1} & \alpha^{(b+1)j_2} & \cdots & \alpha^{(b+1)j_{d-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{(b+n-k-1)(j_1)} & \alpha^{(b+n-k-1)j_2} & \cdots & \alpha^{(b+n-k-1)j_{d-1}} \end{vmatrix} \tag{11.3}$$

$$= \alpha^{b(j_1+j_2+\cdots+j_{d-1})}\det\begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{d-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{j_1} & \alpha^{j_2} & \cdots & \alpha^{j_{d-1}} \end{vmatrix} \tag{11.4}$$

for any columns $j_1, j_2, \ldots, j_{d-1}$. The determinant in 11.3 is called a Vandermonde determinant and is known to have a nonzero determinant for any columns $j_1, j_2, \ldots, j_{d-1}$. If H_{ERS} in 11.2 is considered, any $d-1\times d-1$ determinant of the matrix with columns $j_1, j_2, \ldots, j_{d-1}$ from the submatrix H_{RS} of H_{ERS} is nonzero. There is also a need to cater for determinants that include the appended columns to prove the minimum distance of the code defined by H_{ERS}. Consider the determinant that includes the first appended column of H_{ERS} in $j_1, j_2, \ldots, j_{d-1}$ chosen columns,

$$\det\begin{vmatrix} \alpha^{bj_1} & \alpha^{bj_2} & \cdots & \alpha^{bj_{d-2}} & 1 \\ \alpha^{(b+1)j_1} & \alpha^{(b+1)j_2} & \cdots & \alpha^{(b+1)j_{d-2}} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b+n-k-1)(j_1)} & \alpha^{(b+n-k-1)j_2} & \cdots & \alpha^{(b+n-k-1)j_{d-2}} & 0 \end{vmatrix} \tag{11.5}$$

expanding about the appended column gives Equation 11.5 as,

$$= 1\cdot\det\begin{vmatrix} \alpha^{(b+1)j_1} & \cdots & \alpha^{(b+1)j_{d-2}} \\ \vdots & \ddots & \vdots \\ \alpha^{(b+n-k-1)(j_1)} & \cdots & \alpha^{(b+n-k-1)j_{d-2}} \end{vmatrix}$$

The determinant in 11.5 reduces to a $d-2\times d-2$ determinant of a Vandermonde matrix and is therefore nonzero. This proves that adding the appended column to

H_{ERS} does not decrease the minimum distance from d. Similarly the determinant which includes the second appended column in H_{ERS} reduces to,

$$\det \begin{vmatrix} \alpha^{bj_1} & \cdots & \alpha^{bj_{d-2}} \\ \vdots & \ddots & \vdots \\ \alpha^{(b+n-k-2)(j_1)} & \cdots & \alpha^{(b+n-k-2)j_{d-2}} \end{vmatrix} \tag{11.6}$$

which is also a Vandermonde determinant and is nonzero. Any $d-1 \times d-1$ determinant that contains both of the appended columns will also result in,

$$\det \begin{vmatrix} \alpha^{(b+1)j_1} & \cdots & \alpha^{(b+1)j_{d-2}} \\ \vdots & \ddots & \vdots \\ \alpha^{(b+n-k-2)(j_1)} & \cdots & \alpha^{(b+n-k-2)j_{d-2}} \end{vmatrix} \tag{11.7}$$

after expansion about either of the appended columns. This is also a Vandermonde determinant and is nonzero. Thus every $d-1 \times d-1$ determinant of H_{ERS} is nonzero and the doubly extended RS code has distance d. Notice however that any $d-1 \times d-1$ determinant of the matrix in 11.8,

$$H_{\text{E}} = \begin{bmatrix} 1 & \alpha^{b} & \alpha^{2b} & \cdots & \alpha^{(n-1)b} & 0 \\ 1 & \alpha^{b+1} & \alpha^{2(b+1)} & \cdots & \alpha^{(n-1)(b+1)} & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \alpha^{b+n-k-1} & \alpha^{2(b+n-k-1)} & \cdots & \alpha^{(n-1)(b+n-k-1)} & 0 \end{bmatrix} \tag{11.8}$$

which includes the appended column is as in 11.9,

$$\det \begin{vmatrix} \alpha^{bj_1} & \alpha^{bj_2} & \cdots & \alpha^{bj_{d-2}} & 0 \\ \alpha^{(b+1)j_1} & \alpha^{(b+1)j_2} & \cdots & \alpha^{(b+1)j_{d-2}} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha^{(b+n-k-1)(j_1)} & \alpha^{(b+n-k-1)j_2} & \cdots & \alpha^{(b+n-k-1)j_{d-2}} & 0 \end{vmatrix} \tag{11.9}$$

The determinant in 11.9 after expansion about the appended column reduces to;

$$\det \begin{vmatrix} \alpha^{bj_1} & \cdots & \alpha^{bj_{d-2}} \\ \alpha^{(b+2)j_1} & \cdots & \alpha^{(b+2)j_{d-2}} \\ \vdots & \ddots & \vdots \\ \alpha^{(b+n-k-2)(j_1)} & \cdots & \alpha^{(b+n-k-2)j_{d-2}} \end{vmatrix} \tag{11.10}$$

which is not a Vandermonde determinant[1] therefore is not guaranteed to be nonzero. Appending columns with a 1 in any position except the first and the last, with 0's

[1]Since the row with powers $b+1$ is excluded thus upsetting the sequence b_i, $i \in [0..n-k-1]$.

elsewhere will result in determinants that are not guaranteed to be nonzero. This completes Wolf's (Wolf, 1969) observation. In summary, only the two columns that were appended in 11.2 will guarantee an MDS code from a Vandermonde matrix the only known exception (MacWilliams and Sloane, 1983) being the case where there are 3 rows which allows appending a 3×3 identity matrix[2].

From the previous discussion it can be observed that a Vandermonde matrix has the property that any square submatrix is non-singular (has a nonzero determinant). Another matrix that shares this property with the Vandermonde matrix is the Cauchy matrix (MacWilliams and Sloane, 1983). A Cauchy matrix in a finite field $\mathbb{F}_q$ is formed from two mutually disjoint sets $\{\alpha_1, \alpha_2, \ldots, \alpha_r\}$ and $\{\beta_1, \beta_2, \ldots, \beta_l\}$ and is given by

$$H_{\text{C}} = \begin{bmatrix} \frac{1}{\alpha_1-\beta_1} & \frac{1}{\alpha_1-\beta_2} & \cdots & \frac{1}{\alpha_1-\beta_l} \\ \frac{1}{\alpha_2-\beta_1} & \frac{1}{\alpha_2-\beta_2} & \cdots & \frac{1}{\alpha_2-\beta_l} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{\alpha_r-\beta_1} & \frac{1}{\alpha_r-\beta_2} & \cdots & \frac{1}{\alpha_r-\beta_l} \end{bmatrix} \tag{11.11}$$

such that $r + l \leq |\mathbb{F}_q|$, $\alpha_i, \beta_j \in \mathbb{F}_q$ and $\{\alpha_1, \alpha_2, \ldots, \alpha_r\} \cap \{\beta_1, \beta_2, \ldots, \beta_l\} = \emptyset$. The codes defined by H_{C} are also MDS. Appending a column to H_{C} in the same manner as singly extended RS codes results in a parity check matrix of an extended Cauchy code with $d = r + 1$ of the form,

$$H_{\text{EC}} = \begin{bmatrix} \frac{1}{\alpha_1-\beta_1} & \frac{1}{\alpha_1-\beta_2} & \cdots & \frac{1}{\alpha_1-\beta_l} & 1 \\ \frac{1}{\alpha_2-\beta_1} & \frac{1}{\alpha_2-\beta_2} & \cdots & \frac{1}{\alpha_2-\beta_l} & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{\alpha_r-\beta_1} & \frac{1}{\alpha_r-\beta_2} & \cdots & \frac{1}{\alpha_r-\beta_l} & 0 \end{bmatrix} \tag{11.12}$$

A $d-1 \times d-1$ determinant of H_{EC} with any columns $j_1, j_2, \ldots, j_{d-2}$ and the appended column is given by 11.13,

$$\det \begin{vmatrix} \frac{1}{\alpha_1-\beta_{j_1}} & \frac{1}{\alpha_1-\beta_{j_2}} & \cdots & \frac{1}{\alpha_1-\beta_{j_{d-2}}} & 1 \\ \frac{1}{\alpha_2-\beta_{j_1}} & \frac{1}{\alpha_2-\beta_{j_2}} & \cdots & \frac{1}{\alpha_2-\beta_{j_{d-2}}} & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{1}{\alpha_r-\beta_{j_1}} & \frac{1}{\alpha_r-\beta_{j_2}} & \cdots & \frac{1}{\alpha_r-\beta_{j_{d-2}}} & 0 \end{vmatrix} \tag{11.13}$$

which reduces to 11.14,

$$\det \begin{vmatrix} \frac{1}{\alpha_2-\beta_{j_1}} & \frac{1}{\alpha_2-\beta_{j_2}} & \cdots & \frac{1}{\alpha_2-\beta_{j_{d-2}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{1}{\alpha_r-\beta_{j_1}} & \frac{1}{\alpha_r-\beta_{j_2}} & \cdots & \frac{1}{\alpha_r-\beta_{j_{d-2}}} \end{vmatrix} \tag{11.14}$$

[2]Codes from this construction are called triply extended RS codes.

after expansion about the appended column. Clearly the determinant in 11.14 is a Cauchy determinant formed from the sets $\{\alpha_2,\ldots,\alpha_r\}\cap\{\beta_1,\beta_2,\ldots,\beta_l\}$ and is therefore nonzero. It is also clear that any appended column with a single nonzero entry on any row of H_{EC} will also result in a Cauchy determinant. This is because there is no criterion for ordering of the elements of the two defining sets of the Cauchy matrix i.e. any set of mutually distinct elements will suffice. It is possible append any column with a single nonzero entry to H_{EC} and obtain a Cauchy determinant. In general it is possible to add an identity matrix to the parity check of a Cauchy code and still have minimum distance $d = r+1$. This may appear as a disadvantage for Vandermonde codes when the two codes are compared. However it has to be taken into account that Cauchy codes defined by H_{C} are always in shortened form since the set $\{\beta_1,\beta_2,\ldots,\beta_l\}$ which determines the length of the Cauchy code cannot contain any members of the set $\{\alpha_1,\alpha_2,\ldots,\alpha_r\}$. Appending an identity matrix to H_{C} will make up for the difference in length between the two types of codes. In effect, in terms of code length there is nothing to choose from between the Vandermonde codes and the extended Cauchy codes. In the next section this key difference is taken advantage of so as to extend Goppa codes and obtain new codes.

11.3.2 Construction of Extended Length Goppa Codes

The Goppa polynomial of the form,

$$\acute{G}(x) = x^{r_1}(x-\beta_2)(x-\beta_3)\cdots(x-\beta_\ell) \tag{11.15}$$

is used, with ℓ distinct roots and the set,

$$\acute{L} = \{\mathbb{F}_{q^m} \setminus \{\beta_1,\ldots,\beta_\ell\}\}$$

which is an instance of the Goppa polynomial from Equation (8.4) in Section 8.2 with $r_\mu = 1$ when $\mu > 1$ and with $\beta_1 = 0$. From Equation (8.2) in Section 8.2 the parity check matrix of the $\Gamma(\acute{L},\acute{G})$ code defined by $\acute{G}(x)$ is,

$$\acute{H} = \begin{bmatrix} H_{r_1} \\ H_{r_2} \\ \vdots \\ H_{r_\ell} \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha_1} & \frac{1}{\alpha_2} & \cdots & \frac{1}{\alpha_n} \\ \frac{1}{\alpha_1^2} & \frac{1}{\alpha_2^2} & \cdots & \frac{1}{\alpha_n^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1^{r_1}} & \frac{1}{\alpha_2^{r_1}} & \cdots & \frac{1}{\alpha_n^{r_1}} \\ \frac{1}{\beta_2-\alpha_1} & \frac{1}{\beta_2-\alpha_2} & \cdots & \frac{1}{\beta_2-\alpha_n} \\ \frac{1}{\beta_3-\alpha_1} & \frac{1}{\beta_3-\alpha_2} & \cdots & \frac{1}{\beta_3-\alpha_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\beta_\ell-\alpha_1} & \frac{1}{\beta_\ell-\alpha_2} & \cdots & \frac{1}{\beta_\ell-\alpha_n} \end{bmatrix} \tag{11.16}$$

It is clear from Equation (11.16) and Theorem 3.2 that H_{r_1}, the parity check matrix corresponding to the factor of the Goppa polynomial x^{r_1} from Equation (11.15), is a parity check matrix of a BCH code. The method used to obtain new codes in this thesis involves extending the length of the code by adding columns to the parity check matrix $\acute{H}$ of the code $\Gamma(\acute{L}, \acute{G})$. The matrix $\acute{H}$ is partitioned into two; the parity check matrix of a BCH code defined by x^{r_1} and a Cauchy matrix. The Cauchy matrix with $\ell - 1$ rows is the parity check matrix corresponding to all distinct factors of $\acute{G}(x)$ excluding x^{r_1}.

$$H_{\text{Y}} = \begin{bmatrix} \frac{1}{\alpha_1} & \frac{1}{\alpha_2} & \cdots & \frac{1}{\alpha_n} & 0 & 0 & \cdots & 0 \\ \frac{1}{\alpha_1^2} & \frac{1}{\alpha_2^2} & \cdots & \frac{1}{\alpha_n^2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1^{r_1}} & \frac{1}{\alpha_2^{r_1}} & \cdots & \frac{1}{\alpha_n^{r_1}} & 0 & 0 & \cdots & 0 \\ \frac{1}{\beta_2 - \alpha_1} & \frac{1}{\beta_2 - \alpha_2} & \cdots & \frac{1}{\beta_2 - \alpha_n} & 1 & 0 & \cdots & 0 \\ \frac{1}{\beta_3 - \alpha_1} & \frac{1}{\beta_3 - \alpha_2} & \cdots & \frac{1}{\beta_3 - \alpha_n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\beta_\ell - \alpha_1} & \frac{1}{\beta_\ell - \alpha_2} & \cdots & \frac{1}{\beta_\ell - \alpha_n} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{11.17}$$

The parity check matrix H_{Y} of the lengthened code is given by (11.17). An all-zeros matrix is appended to H_{r_1} while an identity matrix is appended to the Cauchy matrix. In order to represent the matrix H_{Y} with elements from the subfield $\mathbb{F}_q$ the map π_m is applied to elements of the sub-matrix $\acute{H}$ of H_{Y} while the 0's in the appended columns map to $m \times m$ zero matrices and the 1's map to $m \times m$ identity matrices. Each extended symbol of a codeword of $\mathscr{C}_{\text{Y}}$ in $\mathbb{F}_{q^m}$ corresponding to an appended column is therefore an alphabet in $\mathbb{F}_{q^m}$ expanded[3] to an m-tuple in $\mathbb{F}_q$. It is clear that the codes defined by H_{Y} have parameters,

$$\begin{aligned} \text{Length}, n &= |\acute{L}| + m(\ell - 1) \\ \text{Dimension}, k &\geq n - (mr_1 + m(\ell - 1)) \\ &\geq |\acute{L}| - mr_1 \end{aligned} \tag{11.18}$$

11.1 Theorem. *The minimum distance d of the lengthened Goppa code denoted by $\mathscr{C}_{\text{Y}}$ with a parity check matrix H_{Y} is $d \geq \deg(\acute{G}(x)) + 1$.*

Proof. In order to obtain a lower bound on the minimum distance of the code $\mathscr{C}_{\text{Y}}$ it is enough to show that minimum distance of the code defined by H_{Y} in the parent field[4] $\mathbb{F}_{q^m}$ is $d = \deg(\acute{G}(x)) + 1$. This means that the appended columns in H_{Y} do not

[3] This is referred to in literature as subfield image expansion.

[4] Since all subfield subcodes of $\mathscr{C}_{\text{P}}$ have distance at least $\deg(\acute{G}(x)) + 1$.

deteriorate the minimum distance of the code such that it is less than the distance of the code defined by $\acute{H}$.
Recall that a linear code has minimum distance d if any $d-1$ or less columns of its parity check matrix are linearly independent. Alternatively, a code has distance d if any $d-1 \times d-1$ sub-matrix formed from its parity check matrix has a non-zero determinant. Consider Equation (11.17), any $d-1 \times d-1$ matrix (with $d = \deg(\acute{G}(x))+1$) formed from the matrix H_{Y} that does *not* include any of the appended columns is a sub-matrix of $\acute{H}$ (a Vandermonde matrix) and hence has a non-zero determinant. Any $d-1 \times d-1$ sub-matrix of H_{Y} which includes a single appended column having a 1 on a row of H_{Y} corresponding to a root β_μ of $\acute{G}(x)$ for any $\mu > 1$ and zeros elsewhere will have a $d-2 \times d-2$ determinant after expansion about the appended column in question. This $d-2 \times d-2$ determinant is formed from the columns of a parity check matrix of a Goppa code defined with a Goppa polynomial with all the roots of $\acute{G}(x)$ except β_μ and hence has distance $d-1$. The $d-2 \times d-2$ determinant is therefore non-zero. Similarly any $d-1 \times d-1$ sub-matrix of H_{Y} that includes any two of the appended columns having 1's in rows corresponding to roots β_μ and β_ν of $\acute{G}(x)$ with $\nu, \mu > 1$ and zeros elsewhere has a $d-3 \times d-3$ determinant after expansion about the columns in question. This determinant is formed from the columns of a Goppa code with a defining polynomial having all the roots of $\acute{G}(x)$ except β_μ and β_ν. The code has distance $d-2$ therefore the $d-3 \times d-3$ determinant is non-zero. Applying this reasoning successively until all $\ell-1$ appended columns in (11.17) are considered, it can be concluded that any $d-1 \times d-1$ columns of the matrix H_{Y} defined in $\mathbb{F}_{q^m}$ are linearly independent hence the code $\mathscr{C}_{\text{Y}}$ defined in $\mathbb{F}_q$ has distance $d \geq \deg(\acute{G}(x))+1$.

The approach used in the proof of Theorem 11.1 is similar to the one used by Wolf in (Wolf, 1969) to determine the minimum distance of doubly-extended Reed Solomon codes. It is known from (MacWilliams and Sloane, 1983) that Vandermonde and Cauchy matrices both have the property that any square sub-matrix is non-singular. The proof relies on the observation that a Cauchy matrix with r rows may have an $r \times r$ identity matrix appended to it and still retain the property that any $r \times r$ sub-matrix is non-singular. This is not possible with Vandermonde matrices as at most two columns can be appended (as is the case with doubly extended Reed Solomon codes) whilst retaining this property.

11.3.3 Codes with Better Dimensions

From (11.18) it can be observed that the exact dimension of the codes $\mathscr{C}_{\text{Y}}$ depends on the dimension of the BCH code defined by x^{r_1}. The dimension of a BCH code can be completely determined by examining its defining roots. A parity check matrix of a BCH code defined as a Goppa code with polynomial x^{r_1} is given (11.19). By observing (11.19) it is possible to see that if α is a primitive element of $\mathbb{F}_{q^m}$, the defining

roots of the BCH in $\mathbb{F}_q$ consist of the elements of the set $A = \{\alpha^{q^m-2}, \alpha^{q^m-3}, \ldots, \alpha^{q^m-1-r_1}\}$ and all their conjugates.

$$H_{r_1} = \begin{bmatrix} \alpha_1^{(q^m-2)} & \alpha_2^{(q^m-2)} & \cdots & \alpha_n^{(q^m-2)} \\ \alpha_1^{(q^m-3)} & \alpha_2^{(q^m-3)} & \cdots & \alpha_n^{(q^m-3)} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^{(q^m-1-r_1)} & \alpha_2^{(q^m-1-r_1)} & \cdots & \alpha_n^{(q^m-1-r_1)} \end{bmatrix} \tag{11.19}$$

Recall the definition of a conjugacy class. A conjugacy class of an element β of a finite field $\mathbb{F}_{q^m}$ is given as the set,

$$C(\beta) = \{\beta, \beta^q, \beta^{q^2}, ..., \beta^{q^{(e-1)}}\} \quad \beta \in \mathbb{F}_{q^m}$$

where e is the smallest positive integer such that $\beta^{q^e} = \beta$. The set of roots of a BCH code are given by

$$R = \bigcup_{\beta \in A} C(\beta),$$

the codes have redundancy $|R|$ and dimension $k = n - |R|$.

To obtain the best possible dimension it is desirable the BCH code defined by (11.19) to be *narrow sense*. Narrow sense BCH codes tend to have the cardinality $|R|$ to be comparatively small when $|A| = r_1$ is also small. It is possible to shift the roots of the sub-matrix H_{r_1} in $\acute{H}$ from (11.16) so that the BCH code is narrow sense. We can accomplish this by multiplying $\acute{H}$ with a matrix M.

$$M = \begin{bmatrix} \alpha_1^{-(q^m-1-r_1)} & 0 & \cdots & 0 \\ 0 & \alpha_2^{-(q^m-1-r_1)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_n^{-(q^m-1-r_1)} \end{bmatrix}$$

$$\ddot{H} = \acute{H} \times M \tag{11.20}$$

and

$$\ddot{H} = \begin{bmatrix} \alpha_1^{(r_1-1)} & \alpha_2^{(r_1-1)} & \cdots & \alpha_n^{(r_1-1)} \\ \alpha_1^{(r_1-2)} & \alpha_2^{(r_1-2)} & \cdots & \alpha_n^{(r_1-2)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \\ \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_2-\alpha_1} & \frac{\alpha_2^{-(q^m-1-r_1)}}{\beta_2-\alpha_2} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_2-\alpha_n} \\ \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_3-\alpha_1} & \frac{\alpha_2^{-(q^m-1-r_1)}}{\beta_3-\alpha_2} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_3-\alpha_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_1} & \frac{\alpha_2^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_2} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_n} \end{bmatrix} \tag{11.21}$$

11.2 *Theorem.* *The code defined by $\ddot{H}$ restricted to $\mathbb{F}_q$ has minimum distance $d \geq \deg(\acute{G}(x)) + 1$.*

Proof. Recall the definition of generalised Reed Solomon (GRS) codes from (MacWilliams and Sloane, 1983). A GRS code, denoted by $\mathrm{GRS}_k(\alpha, \mathbf{v})$, consists of all the vectors,

$$(v_1 F(\alpha_1), v_2 F(\alpha_2), \ldots, v_n F(\alpha_n)))$$

where $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ consists of distinct elements of $\mathbb{F}_{q^m}$, a template $\mathbf{v} = (v_1, v_2, \ldots, v_n)$ consists of arbitrary elements from $\mathbb{F}_{q^m}$ none of which is zero and $F(x)$ is a polynomial of degree at most $k-1$. Also from (MacWilliams and Sloane, 1983) it is known that Goppa codes defined by some $G(x)$ of degree r and the set $L = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ are subfield subcodes of $\mathrm{GRS}_{n-r}(\alpha, \mathbf{v})$ with $k = n - r$ and,

$$v_i = \frac{G(\alpha_i)}{\prod_{j \neq i}(\alpha_i - \alpha_j)}, \quad i = 1, \ldots, n. \tag{11.22}$$

Again from (MacWilliams and Sloane, 1983) it is observed that dual code of a $\mathrm{GRS}_k(\alpha, \mathbf{v})$ code is also a GRS code of the form $\mathrm{GRS}_{n-k}(\alpha, \acute{\mathbf{v}})$ for some template $\acute{\mathbf{v}}$. Clearly the code defined by $\ddot{H}$ in $\mathbb{F}_{q^m}$ is a GRS code of the form $\mathrm{GRS}_{n-\deg(\acute{G}(x))}(\alpha, \mathbf{uv})$ where $\mathbf{v}$ is defined as in (11.22) and

$$\acute{\mathbf{u}} = (\alpha_1^{-(q^m-1-r_1)}, \alpha_2^{-(q^m-1-r_1)}, \ldots, \alpha_n^{-(q^m-1-r_1)}).$$

Since the code defined by $\ddot{H}$ in $\mathbb{F}_q$ is a subfield subcode it has minimum distance at least that of the code defined in $\mathbb{F}_{q^m}$. GRS codes are maximum distance separable and in this case the code defined in $\mathbb{F}_{q^m}$ has distance $\deg(\acute{G}(x)) + 1$, therefore its subfield subcode has distance $d \geq \deg(\acute{G}(x)) + 1$. ∎

Codes defined by $\ddot{H}$ in $\mathbb{F}_q$, denoted by $\ddot{\mathscr{C}}_{\text{Y}}$, have a better dimension but the same minimum distance was the code defined by $\acute{H}$. It is possible to add columns to the parity-check matrix of the code defined by $\ddot{H}$ in $\mathbb{F}_q$ in the same manner as in (11.17). Equation (11.27) shows the parity check matrix of the lengthened code. As with codes defined by H_{Y} the length, dimension and minimum distance of the codes are,

$$\begin{aligned} \text{Length}, n &= |\acute{L}| + m(\ell - 1) \\ \text{Dimension}, k &\geq |\acute{L}| - mr_1 \\ \text{Distance}, d &\geq \deg(\acute{G}(x)) + 1 \end{aligned} \tag{11.23}$$

$$\ddot{H}_{\text{Y}} = \begin{bmatrix} \alpha_1^{(r_1-1)} & \cdots & \alpha_n^{(r_1-1)} & 0 & \ldots & 0 \\ \alpha_1^{(r_1-2)} & \cdots & \alpha_n^{(r_1-2)} & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & \cdots & 1 & 0 & \ldots & 0 \\ & & & & & \\ \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_2-\alpha_1} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_2-\alpha_n} & 1 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_1} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_n} & 0 & \ldots & 1 \end{bmatrix} \tag{11.24}$$

It is straight forward to show that codes defined by $\ddot{H}_{\text{Y}}$ have distance at least $\deg(\acute{G}(x)) + 1$ by using the same reasoning as in Theorem 11.1.

11.3.4 An Example

As an illustration of the construction, a polynomial $\acute{G}(x) = x^2(x+1)$ with coefficients from $\mathbb{F}_8$ is used to define an extended Goppa code in $\mathbb{F}_2$. The finite field $\mathbb{F}_8$ is defined with the primitive polynomial $s^3 + s + 1$ and has α as a primitive element. The set $\acute{L}$ corresponding to $\acute{G}(x)$ is then given by,

$$\acute{L} = \begin{bmatrix} \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \end{bmatrix}$$

and from (11.16) the matrix $\acute{H}$ is,

$$\acute{H} = \begin{bmatrix} \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha \\ \alpha^5 & \alpha^3 & \alpha & \alpha^6 & \alpha^4 & \alpha^2 \\ \alpha^4 & \alpha & \alpha^6 & \alpha^2 & \alpha^3 & \alpha^5 \end{bmatrix}$$

For this example clearly $\ell = 2$, $r_1 = 2$, $|\acute{L}| = 6$ and $m = 3$. To make the BCH part of $\acute{H}$ narrow sense it is multiplied with the matrix M,

$$M = \begin{bmatrix} \alpha^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha^6 & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha^3 & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha^5 \end{bmatrix}$$

and this results in the matrix $\ddot{H}$,

$$\ddot{H} = \acute{H} \times M = \begin{bmatrix} \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ \alpha^6 & \alpha^5 & \alpha^5 & \alpha^3 & \alpha^6 & \alpha^3 \end{bmatrix}$$

adding a single column the matrix $\ddot{H}_{\text{Y}}$ in $\mathbb{F}_8$ is obtained.

$$\ddot{H}_{\text{Y}} = \begin{bmatrix} \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ \alpha^6 & \alpha^5 & \alpha^5 & \alpha^3 & \alpha^6 & \alpha^3 & 1 \end{bmatrix}$$

$\ddot{H}_{\text{Y}}$ is then expressed in $\mathbb{F}_2$ as,

$$\ddot{H}_{\text{Y}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The matrix $\ddot{H}_{\text{Y}}$ is the parity check matrix of the extended length Goppa code and

after row reductions the code has parameters $[9,2,5]_2$.

$$\ddot{H}_{\text{Y}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

It can be observed that the minimum distance of the code is one more than the lower bound i.e. $\deg(\acute{G}(x))+1 = 4$. Results on codes obtained from the construction method are presented. These codes have minimum distances better than the codes in (Grassl, 2007) with same length and rate. They are derived from Goppa codes defined by the polynomial in (11.15). Goppa polynomials with coefficients in $\mathbb{F}_{q^m}$ of the form,

$$G(x) = x^{r_1} \prod_{i=0}^{\ell-2} (x - \alpha^i)$$

where α is the primitive element of $\mathbb{F}_{q^m}$ are used. Tables 11.1 - 11.3 give a summary of the results of new codes in $\mathbb{F}_7$, $\mathbb{F}_8$ and $\mathbb{F}_9$ which are subfield subcodes of codes defined in $\mathbb{F}_{49}$, $\mathbb{F}_{64}$, and $\mathbb{F}_{81}$ respectively. The codes are represented in the form $[n,k,d]_q$. The dimensions of the codes are confirmed by expressing their parity check matrices in reduced echelon form. It is worth noting that some of the codes in Tables 11.1 - 11.3 can be obtained from other codes in the tables by shortening in one or more positions. For example $\mathscr{C}_{37}$ can be obtained from $\mathscr{C}_{38}$ by shortening in one position.

0 #	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_1$	49	2	2	5	$[52,41,7]_7$	$[52,41,6]_7$
$\mathscr{C}_2$	49	2	9	6	$[53,29,15]_7$	$[53,29,14]_7$

Table 11.1: New Codes in $\mathbb{F}_7$

#	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_3$	64	2	1	9	$[71,54,10]_8$	$[71,54,9]_8$
$\mathscr{C}_4$	64	2	1	12	$[74,51,13]_8$	$[74,51,12]_8$
$\mathscr{C}_5$	64	2	1	13	$[75,50,14]_8$	$[75,50,13]_8$
$\mathscr{C}_6$	64	2	10	6	$[68,42,16]_8$	$[68,42,15]_8$
$\mathscr{C}_7$	64	2	10	7	$[69,41,17]_8$	$[69,41,16]_8$
$\mathscr{C}_8$	64	2	11	6	$[68,40,17]_8$	$[68,40,16]_8$

Table 11.2: New Codes in $\mathbb{F}_8$

#	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_9$	81	2	1	6	$[85,74,7]_9$	$[85,74,6]_9$
$\mathscr{C}_{10}$	81	2	2	6	$[85,72,8]_9$	$[85,72,7]_9$
$\mathscr{C}_{11}$	81	2	3	6	$[85,70,9]_9$	$[85,70,8]_9$
$\mathscr{C}_{12}$	81	2	4	6	$[85,68,10]_9$	$[85,69,9]_9$
$\mathscr{C}_{13}$	81	2	11	6	$[85,57,17]_9$	$[85,57,16]_9$
$\mathscr{C}_{14}$	81	2	12	6	$[85,55,18]_9$	$[85,55,17]_9$
$\mathscr{C}_{15}$	81	2	21	6	$[85,42,27]_9$	$[85,42,26]_9$
$\mathscr{C}_{16}$	81	2	22	6	$[85,40,28]_9$	$[85,40,27]_9$
$\mathscr{C}_{17}$	81	2	1	7	$[86,73,8]_9$	$[86,73,7]_9$
$\mathscr{C}_{18}$	81	2	2	7	$[86,71,9]_9$	$[86,71,8]_9$
$\mathscr{C}_{19}$	81	2	3	7	$[86,69,10]_9$	$[86,69,9]_9$
$\mathscr{C}_{20}$	81	2	11	7	$[86,56,18]_9$	$[86,56,17]_9$
$\mathscr{C}_{21}$	81	2	12	7	$[86,54,19]_9$	$[86,54,18]_9$
$\mathscr{C}_{22}$	81	2	21	7	$[86,41,28]_9$	$[86,41,27]_9$
$\mathscr{C}_{23}$	81	2	2	8	$[87,70,10]_9$	$[87,70,9]_9$
$\mathscr{C}_{24}$	81	2	11	8	$[87,55,19]_9$	$[87,55,18]_9$
$\mathscr{C}_{25}$	81	2	1	9	$[88,71,10]_9$	$[88,71,9]_9$
$\mathscr{C}_{26}$	81	2	2	9	$[88,69,11]_9$	$[88,69,10]_9$
$\mathscr{C}_{27}$	81	2	10	9	$[88,55,19]_9$	$[88,55,18]_9$
$\mathscr{C}_{28}$	81	2	11	9	$[88,54,20]_9$	$[88,54,19]_9$
$\mathscr{C}_{29}$	81	2	1	10	$[89,70,11]_9$	$[89,70,10]_9$
$\mathscr{C}_{30}$	81	2	1	11	$[90,69,12]_9$	$[90,69,11]_9$
$\mathscr{C}_{31}$	81	2	1	12	$[91,68,13]_9$	$[91,68,12]_9$
$\mathscr{C}_{32}$	81	2	2	12	$[91,66,14]_9$	$[91,66,13]_9$
$\mathscr{C}_{33}$	81	2	1	13	$[92,67,14]_9$	$[92,67,13]_9$
$\mathscr{C}_{34}$	81	2	2	13	$[92,65,15]_9$	$[92,65,14]_9$
$\mathscr{C}_{35}$	81	2	1	15	$[94,65,16]_9$	$[94,65,15]_9$
$\mathscr{C}_{36}$	81	2	2	15	$[94,63,17]_9$	$[94,63,16]_9$
$\mathscr{C}_{37}$	81	2	2	13	$[92,65,15]_9$	$[92,65,14]_9$
$\mathscr{C}_{38}$	81	2	1	14	$[93,66,15]_9$	$[93,66,14]_9$
$\mathscr{C}_{39}$	81	2	1	15	$[94,65,16]_9$	$[94,65,15]_9$
$\mathscr{C}_{40}$	81	2	2	15	$[94,63,17]_9$	$[94,63,16]_9$

Table 11.3: New Codes in $\mathbb{F}_9$

11.3.5 Adding One More Column

It is possible to add a single column to the parity-check matrix H_{Y} in the same manner as singly extended Reed Solomon codes (Wolf, 1969). A parity-check matrix H_{P},

$$H_{\text{P}} = \begin{bmatrix} 0 & \frac{1}{\alpha_1} & \cdots & \frac{1}{\alpha_n} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\alpha_1^2} & \cdots & \frac{1}{\alpha_n^2} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \frac{1}{\alpha_1^{r_1}} & \cdots & \frac{1}{\alpha_n^{r_1}} & 0 & 0 & \cdots & 0 \\ 0 & \frac{1}{\beta_2-\alpha_1} & \cdots & \frac{1}{\beta_2-\alpha_n} & 1 & 0 & \cdots & 0 \\ 0 & \frac{1}{\beta_3-\alpha_1} & \cdots & \frac{1}{\beta_3-\alpha_n} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{1}{\beta_\ell-\alpha_1} & \cdots & \frac{1}{\beta_\ell-\alpha_n} & 0 & 0 & \cdots & 1 \end{bmatrix} \tag{11.25}$$

is formed which leads to the next theorem.

11.3 Theorem. *The linear code $\mathscr{C}_P$ defined by the parity-check matrix H_P in $\mathbb{F}_q$ has*

$$\begin{aligned} \textit{Length} &= n+1 \\ \textit{Dimension} &= k+1 \\ \textit{Distance} &= d \end{aligned}$$

where $\mathscr{C}_{\text{Y}}$ is a code of length n, dimension k and distance d defined by the parity-check matrix H_{Y} from (11.17) .

Proof. As in Theorem 11.1 it is sufficient to prove that appending the column does not cause the distance of $\mathscr{C}_{\text{P}}$ to be less than that of $\mathscr{C}_{\text{Y}}$. The length and dimension of the codes $\mathscr{C}_{\text{P}}$ are 1 more than that of $\mathscr{C}_{\text{Y}}$ since the parity check matrix H_{P} has an additional column.

Since H_{Y} has already been established in Theorem 11.1 to have all $d-1$ columns linearly independent and thus $\mathscr{C}_{\text{Y}}$ has minimum distance $d \geq \deg(\acute{G}(x))+1$, the case where $d-1$ random chosen columns of H_{P} include the new appended column is considered. The determinant of a $d-1 \times d-1$ submatrix of H_{P} that includes the new appended column can be found by expanding about the column in question. This $d-1 \times d-1$ determinant of H_{P} which includes the appended column will have a 1 on a row of H_{P} corresponding to the defining root of the BCH code α^{r_1} i.e. the last row of the Vandermonde matrix defined by x^{r_1} and is zero elsewhere thus forming a $d-2 \times d-2$ determinant. This $d-2 \times d-2$ determinant is formed from the columns of a parity check matrix of a Goppa code defined with a Goppa polynomial defined

by

$$x^{r_1-1}(x-\beta_2)\cdots(x-\beta_\ell) \tag{11.26}$$

and hence has a non-zero determinant. All $d-1\times d-1$ determinants of H_{P} are non-zero, therefore $\mathscr{C}_{\text{P}}$ defined in $\mathbb{F}_q$ has minimum distance $d \geq \deg(\acute{G}(x))+1$. ∎

Every $\mathscr{C}_{\text{Y}}$ is a code shortened from $\mathscr{C}_{\text{P}}$ in one coordinate. It is possible to shift the roots of the BCH part of the matrix H_{P} in the same way as (11.20) so that better dimensions are obtained by multiplying H_{P} with a matrix M,

$$M = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & \alpha_1^{-(q^m-1-r_1)} & 0 & \cdots & 0 \\ 0 & 0 & \alpha_2^{-(q^m-1-r_1)} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \alpha_n^{-(q^m-1-r_1)} \end{bmatrix}$$

corresponding to a template,

$$\mathbf{v} = \{1, \alpha_1^{-(q^m-1-r_1)}, \alpha_2^{-(q^m-1-r_1)}, \ldots, \alpha_n^{-(q^m-1-r_1)}\}$$

so that $\ddot{H}_{\text{P}} = H_{\text{P}} \times M$ which results in,

$$\ddot{H}_{\text{P}} = \begin{bmatrix} 0 & \alpha_1^{(r_1-1)} & \cdots & \alpha_n^{(r_1-1)} & 0 & \ldots & 0 \\ 0 & \alpha_1^{(r_1-2)} & \cdots & \alpha_n^{(r_1-2)} & 0 & \ldots & 0 \\ 0 & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 & 0 & \ldots & 0 \\ & & & & & & \\ 0 & \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_2-\alpha_1} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_2-\alpha_n} & 1 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \frac{\alpha_1^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_1} & \cdots & \frac{\alpha_n^{-(q^m-1-r_1)}}{\beta_\ell-\alpha_n} & 0 & \ldots & 1 \end{bmatrix} \tag{11.27}$$

It is clear that one can easily choose the template v so that the BCH code is defined with any set of roots. Let $\ddot{\mathscr{C}}_{\text{P}}$ denote the code defined by $\ddot{H}_{\text{P}}$. Clearly, Theorem 11.3 describes the parameters of $\ddot{\mathscr{C}}_{\text{P}}$. New codes obtained from this construction of the form $\ddot{\mathscr{C}}_{\text{P}}$ are now given. Tables 11.4 - 11.6 give new codes of the form $\ddot{\mathscr{C}}_{\text{P}}$. As with codes obtained previously the dimension of each of the codes is confirmed using row reductions while their minimum distances are obtained from the bound in Theorem 11.3.

#	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_{41}$	49	2	2	5	$[53,42,7]_7$	$[53,42,6]_7$
$\mathscr{C}_{42}$	49	2	9	5	$[53,31,14]_7$	$[53,31,13]_7$
$\mathscr{C}_{43}$	49	2	17	5	$[53,20,22]_7$	$[53,20,21]_7$
$\mathscr{C}_{44}$	49	2	1	6	$[54,43,7]_7$	$[54,43,6]_7$
$\mathscr{C}_{45}$	49	2	9	6	$[54,30,15]_7$	$[54,30,14]_7$

Table 11.4: New Codes $\ddot{\mathscr{C}}_P$ in $\mathbb{F}_7$

#	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_{46}$	64	2	10	5	$[68,44,15]_8$	$[68,44,14]_8$
$\mathscr{C}_{47}$	64	2	12	5	$[68,40,17]_8$	$[68,40,16]_8$
$\mathscr{C}_{48}$	64	2	19	5	$[68,31,24]_8$	$[68,31,23]_8$
$\mathscr{C}_{49}$	64	2	20	5	$[68,29,25]_8$	$[68,29,24]_8$
$\mathscr{C}_{50}$	64	2	10	6	$[68,43,16]_8$	$[68,43,15]_8$
$\mathscr{C}_{51}$	64	2	19	6	$[69,30,25]_8$	$[69,30,24]_8$
$\mathscr{C}_{52}$	64	2	10	7	$[70,42,17]_8$	$[70,42,16]_8$
$\mathscr{C}_{53}$	64	2	10	8	$[71,41,18]_8$	$[71,41,17]_8$
$\mathscr{C}_{54}$	64	2	1	9	$[72,55,10]_8$	$[72,55,10]_8$
$\mathscr{C}_{55}$	64	2	1	12	$[75,52,13]_8$	$[75,53,12]_8$
$\mathscr{C}_{56}$	64	2	1	13	$[76,51,14]_8$	$[76,51,13]_8$

Table 11.5: New Codes $\ddot{\mathscr{C}}_P$ in $\mathbb{F}_8$

#	q^m	m	r_1	ℓ	Codes	Codes in (Grassl, 2007)
$\mathscr{C}_{57}$	81	2	11	5	$[85,59,16]_9$	$[85,59,15]_9$
$\mathscr{C}_{58}$	81	2	21	5	$[85,44,26]_9$	$[85,44,25]_9$
$\mathscr{C}_{59}$	81	2	31	5	$[85,31,36]_9$	$[85,31,35]_9$
$\mathscr{C}_{60}$	81	2	1	6	$[86,75,7]_9$	$[86,75,6]_9$
$\mathscr{C}_{61}$	81	2	11	6	$[86,58,17]_9$	$[86,58,16]_9$
$\mathscr{C}_{62}$	81	2	21	6	$[86,43,27]_9$	$[86,43,26]_9$
$\mathscr{C}_{63}$	81	2	1	7	$[87,74,8]_9$	$[87,74,7]_9$
$\mathscr{C}_{64}$	81	2	11	7	$[87,57,18]_9$	$[87,57,17]_9$
$\mathscr{C}_{65}$	81	2	21	7	$[87,42,28]_9$	$[87,42,27]_9$
$\mathscr{C}_{66}$	81	2	1	8	$[88,73,9]_9$	$[88,73,8]_9$
$\mathscr{C}_{67}$	81	2	11	8	$[88,56,19]_9$	$[88,56,18]_9$
$\mathscr{C}_{68}$	81	2	21	8	$[88,41,29]_9$	$[88,41,28]_9$
$\mathscr{C}_{69}$	81	2	1	9	$[89,72,10]_9$	$[89,72,9]_9$
$\mathscr{C}_{70}$	81	2	11	9	$[89,55,20]_9$	$[89,55,19]_9$
$\mathscr{C}_{71}$	81	2	1	10	$[90,71,11]_9$	$[90,71,10]_9$
$\mathscr{C}_{72}$	81	2	1	11	$[91,70,12]_9$	$[91,70,11]_9$
$\mathscr{C}_{73}$	81	2	1	12	$[92,69,13]_9$	$[92,69,12]_9$
$\mathscr{C}_{74}$	81	2	1	13	$[93,68,14]_9$	$[93,68,13]_9$
$\mathscr{C}_{75}$	81	2	1	14	$[94,67,15]_9$	$[94,67,14]_9$
$\mathscr{C}_{76}$	81	2	1	15	$[95,66,16]_9$	$[95,66,15]_9$
$\mathscr{C}_{77}$	81	2	1	16	$[96,65,17]_9$	$[96,65,16]_9$

Table 11.6: New Codes $\ddot{\mathscr{C}}_P$ in $\mathbb{F}_9$

11.4 Nested Structure and Construction X

It is possible to form a chain of nested codes from this construction. A linear code $\mathscr{C}_1$ is a subcode of a linear code $\mathscr{C}_2$ if all the codewords of $\mathscr{C}_1$ are contained in $\mathscr{C}_2$. Suppose these codes have parity check matrices H_1 and H_2 respectively, then the vector space defined by H_1 contains the vector space defined by H_2.

$$\mathscr{C}_1 \subset \mathscr{C}_2 \text{ iff } H_2 \subset H_1$$

The difference between the dimensions of a code and its subcode is called co-dimension.

11.4 Theorem. *Any extended Goppa code $\Gamma(\acute{L}, \acute{G}_1)$ defined by a Goppa polynomial,*

$$\acute{G}_1(x) = x^s(x-\beta_2)(x-\beta_3)\cdots(x-\beta_\ell)$$

is a subcode (of co-dimension $t-s$) of the Goppa code $\Gamma(\acute{L}, \acute{G}_2)$ defined by the Goppa polynomial,

$$\acute{G}_2(x) = x^t(x-\beta_2)(x-\beta_3)\cdots(x-\beta_\ell)$$

provided $t < s$ and $\beta_i \in \mathbb{F}_{q^m}$.

Proof. Implied in Theorem 11.4, $\acute{G}_1$ and $\acute{G}_2$ must have exactly the same roots not necessarily with the same multiplicity. If H_{Y_1} is parity check matrix of the extended Goppa code defined by $\acute{G}_1$ and H_{Y_2} is the parity check matrix of the extended Goppa code defined by $\acute{G}_2$ then from Equation 11.17 it is clear that,

$$H_{Y_2} \subset H_{Y_1}$$

thus $\mathscr{C}_{Y_1} \subset \mathscr{C}_{Y_2}$. ∎

11.1 Corollary. *Any code $\ddot{\mathscr{C}}_{Y_1}$ derived from a Goppa code with polynomial $\acute{G}_1$ is subcode of the code $\ddot{\mathscr{C}}_{Y_2}$ derived from a Goppa code with polynomial $\acute{G}_2$ provided both codes are defined with the same template $\mathbf{v} = (v_1, v_2, \ldots, v_n)$.*

Clearly if codes $\ddot{\mathscr{C}}_{Y_1}$ and $\ddot{\mathscr{C}}_{Y_2}$ have different templates then the codewords of $\ddot{\mathscr{C}}_{Y_1}$ will not satisfy the parity check equations of $\ddot{\mathscr{C}}_{Y_2}$. Corollary 11.1 leads to a well known method of extending linear codes, Construction X as presented in Theorem 2.1. Tables 11.7 - 11.9 give new codes obtained from construction X with complete

information on the codes $\mathscr{C}_2$, their corresponding subcodes $\mathscr{C}_1$ and auxiliary codes. The template $\mathbf{v}$ is fixed for codes $\mathscr{C}_1$ and $\mathscr{C}_2$ such that,

$$\mathbf{v} = \{\alpha_1^{-(q^m-2)}, \alpha_2^{-(q^m-2)}, \ldots, \alpha_n^{-(q^m-2)}\}$$

#	$\mathscr{C}_2$			$\mathscr{C}_1$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{78}$	9	5	$[54,31,14]_7$	11	5	$[58,28,16]_7$	$[4,3,2]_7$	$[58,31,16]_7$	$[53,31,15]_7$

Table 11.7: New Codes From Construction X in $\mathbb{F}_7$

#	$\mathscr{C}_2$			$\mathscr{C}_1$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{79}$	10	4	$[66,44,14]_8$	12	4	$[66,40,16]_8$	$[5,4,2]_8$	$[71,44,16]_8$	$[71,44,15]_8$
$\mathscr{C}_{80}$	11	4	$[66,42,15]_8$	13	4	$[66,38,17]_8$	$[5,4,2]_8$	$[71,42,17]_8$	$[71,42,16]_8$
$\mathscr{C}_{81}$	10	4	$[66,44,14]_8$	12	4	$[66,40,16]_8$	$[5,4,2]_8$	$[71,44,16]_8$	$[71,44,15]_8$
$\mathscr{C}_{82}$	19	4	$[66,31,23]_8$	21	4	$[66,27,25]_8$	$[5,4,2]_8$	$[71,31,25]_8$	$[71,31,24]_8$

Table 11.8: New Codes From Construction X in $\mathbb{F}_8$

#	$\mathscr{C}_2$			$\mathscr{C}_1$			Auxiliary codes	New Codes	Codes in (Grassl, 2007)
	r_1	ℓ	Codes	r_1	ℓ	Codes			
$\mathscr{C}_{83}$	11	4	$[83,59,15]_9$	13	4	$[83,55,17]_9$	$[5,4,2]_9$	$[88,59,17]_9$	$[88,59,16]_9$
$\mathscr{C}_{84}$	11	4	$[83,59,15]_9$	14	4	$[83,53,18]_9$	$[8,6,3]_9$	$[91,59,18]_9$	$[91,59,17]_9$
$\mathscr{C}_{85}$	12	4	$[83,57,16]_9$	14	4	$[83,53,18]_9$	$[5,4,2]_9$	$[88,57,18]_9$	$[88,57,17]_9$
$\mathscr{C}_{86}$	12	4	$[83,57,16]_9$	15	4	$[83,51,19]_9$	$[8,6,3]_9$	$[91,57,19]_9$	$[91,57,18]_9$
$\mathscr{C}_{87}$	13	4	$[83,55,17]_9$	15	4	$[83,51,19]_9$	$[5,4,2]_9$	$[88,55,19]_9$	$[88,55,18]_9$
$\mathscr{C}_{88}$	21	4	$[83,44,25]_9$	23	4	$[83,40,27]_9$	$[5,4,2]_9$	$[88,44,27]_9$	$[88,44,26]_9$
$\mathscr{C}_{89}$	21	4	$[83,44,25]_9$	24	4	$[83,38,28]_9$	$[8,6,3]_9$	$[91,44,28]_9$	$[91,44,27]_9$
$\mathscr{C}_{90}$	22	4	$[83,42,26]_9$	24	4	$[83,38,28]_9$	$[5,4,2]_9$	$[88,42,28]_9$	$[88,42,27]_9$
$\mathscr{C}_{91}$	11	5	$[84,58,16]_9$	13	5	$[84,54,18]_9$	$[5,4,2]_9$	$[89,58,18]_9$	$[89,58,17]_9$
$\mathscr{C}_{92}$	11	5	$[84,58,16]_9$	14	5	$[84,52,19]_9$	$[8,6,3]_9$	$[92,58,19]_9$	$[92,58,18]_9$
$\mathscr{C}_{93}$	12	5	$[84,56,17]_9$	14	5	$[84,52,19]_9$	$[5,4,2]_9$	$[89,56,19]_9$	$[89,56,18]_9$
$\mathscr{C}_{94}$	21	5	$[84,43,26]_9$	23	5	$[84,39,28]_9$	$[5,4,2]_9$	$[89,43,28]_9$	$[89,43,27]_9$
$\mathscr{C}_{95}$	11	6	$[85,57,17]_9$	13	6	$[85,53,19]_9$	$[5,4,2]_9$	$[90,57,19]_9$	$[90,57,18]_9$
$\mathscr{C}_{96}$	11	7	$[86,56,18]_9$	13	7	$[86,52,20]_9$	$[5,4,2]_9$	$[91,56,20]_9$	$[91,56,19]_9$

Table 11.9: New Codes From Construction X in $\mathbb{F}_9$

11.5 Summary

An alternative construction of extended Goppa codes is given. The first construction in Chapter 8 uses the concept of a modified Goppa code (Goppa, 1971) and thus produces longer codes than the method in this Chapter. The method presented in this Chapter utilises the fact that Goppa codes defined in the parent field are GRS codes. Since the codes are extensions of BCH codes, one can choose a template that extends any type of BCH code. Both methods are generalisations of Construction P by Sugiyama et al. (1976).

12. SUMMARY AND FUTURE RESEARCH

12.1 Summary and Contributions

Research has focused mainly on finding improved algebraic codes with better properties than any known codes. Two important classes of codes invented by V.D Goppa namely Goppa codes and AG codes feature prominently. AG codes were studied in-depth and their advantages and benefits in comparison to some best known codes are investigated. Decoding performance of AG and nonbinary BCH codes is compared in the AWGN using soft and hard decision decoding and erasure channels using maximum likelihood decoding. The BMSA decoding was presented for the bounded distance decoding of AG codes while the classic BMA was presented for BCH codes for transmission in the AWGN channel. Symbol based ordered reliability decoding was carried out for soft decision decoding in the AWGN channel for both AG and BCH codes. Finally maximum-likelihood erasure decoding (in-place) is presented for decoding these codes in the erasure channel. Soft and hard decision performance in the AWGN channel shows that the codes have similar performance. In the erasure channel AG codes show superior performance at low probabilities of erasure, an indication of a more favourable low weight distribution. Research naturally lead to finding improved codes from AG codes. A construction of generalised AG codes that utilised places of degree larger than one and a concatenation concept was presented. Using this method 237 codes in the finite field $\mathbb{F}_{16}$ from four curves with better minimum distances than any known codes were presented. Many improvements on constructible codes were also presented. The search for new codes was then extended to Goppa codes. A construction of extended binary Goppa codes was generalised to nonbinary codes. The concept of an extended Goppa code was used to obtain improved codes. In total 48 new codes in finite fields $\mathbb{F}_7$, $\mathbb{F}_8$ and $\mathbb{F}_9$ were presented directly from this method. Using construction X, 30 further improvements were also obtained. More improvements are also possible from simple modifications of the obtained codes. Finally an alternative method of obtaining these codes is given. A method of shortening linear codes whilst increasing the minimum distance is analysed and generalised. The method works by examining the low weight codewords of a code. A link between this shortening method and meth-

ods of extending codes was presented. Codes with a special structure from Goppa codes are used to obtain 4 new binary codes. A method of extending BCH codes was then presented. The method is shown to be as good as an optimal method of constructing codes; construction X, in cases when the field size is small. The method provides an insight into extending algebraic codes and can be used to obtain best known codes. The major contributions are summarised below,

- Algebraic geometry codes are studied in great detail with special attention given to their construction and decoding. The practical performance of these codes is evaluated and compared with previously known codes in different communication channels.
- Furthermore many new codes that have better minimum distance than the best known codes with the same length and dimension are presented from a generalised construction of algebraic geometry codes.
- A construction of binary extended Goppa codes is generalised to codes with nonbinary alphabets and as a result many new codes are found. This construction is shown as an efficient way to extend another well known class of algebraic codes, BCH codes.
- A generic method of shortening codes whilst increasing the minimum distance is generalised. An analysis of this method reveals a close relationship with methods of extending codes. Some new codes from Goppa codes are found by exploiting this relationship.
- Finally an extension method for BCH codes is presented and this method is shown be as good as a well known method of code extension in certain cases.

12.2 Future Research Directions

Xing's generalised construction of algebraic geometry codes holds promise of producing new codes. However this hinges on finding constructible curves with many places of small degree and small genera. Further contributions to the tables in (Schimd and Shurer, 2004) from generalised AG codes are possible for larger fields if one scours the literature for constructible curves with good genera. The extension method for Goppa which is a generalisation of construction P for nonbinary codes can be extended to other alternant codes like Srivastava codes. A thorough search may well reveal some new codes. An extensive search of the database of best known codes in MAGMA for improved codes from shortening can be carried out if sufficient computational resources are available [1]. If best known codes by shortening are found it may be useful not to rely on the lower bound (as the distance may

[1]The MAGMA software used in this research was issued under a student license and as such has limited memory capacity.

be larger) on minimum distance of codes produced by shortening instead one can find the minimum distance by brute force. In a more general context, a method of constructing algebraic codes with bad dual codes is an appealing area of research. These type of codes can be used to construct good LDPC codes as not much is known on the structural properties of LDPC codes aside from their sparseness.

Part V

Back Matter

13. APPENDIX

13.1 Construction P For Binary Goppa Codes From (Sugiyama et al., 1976)

Let $G(x)$ be a polynomial of degree $2t$ with coefficients from $\mathbb{F}_{2^m}$ such that the factorisation over $\mathbb{F}_{2^m}[x]$ is given by

$$G(x) = \prod_{u=1}^{s_0} (x-\beta_u)^{b_u} \prod_{u=s_0+1}^{s} (g_u(x))^{b_u}, \tag{13.1}$$

where β_u, $u = 1,\ldots,s_0$ are distinct elements of $\mathbb{F}_{2^m}$; b_u, $u = 1,\ldots,s$ are even integers; and s_0 satisfies $0 \le s_0 \le t$. In addition, for $u = s_0+1,\ldots,s$ the roots of the irreducible factors $g_u(x)$ are not in $\mathbb{F}_{2^m}$. Let $L = \{\alpha_1,\ldots,\alpha_n\}$ denote the elements of $\mathbb{F}_{2^m}$ that are not roots of the Goppa polynomial $G(x)$. The parity check matrix of the code $\mathscr{C}_P$ is given by

$$H_{P[L,G]} = \begin{bmatrix} H_0 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ H_1 & H_I & \mathbf{0} & \cdots & \mathbf{0} \\ H_2 & \mathbf{0} & H_I & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ H_{s_0} & \mathbf{0} & \mathbf{0} & \cdots & H_I \\ \mathbf{0} & H_J & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & H_J & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & H_J \end{bmatrix}. \tag{13.2}$$

Here H_0 is the parity check matrix of the Goppa code defined by the polynomial

$$\prod_{u=1}^{s_0} (x-\beta_u)^{\max\{b_u-3,0\}} \prod_{u=s_0+1}^{s} (g_u(x))^{b_u-1}.$$

The matrices H_u, $u = 1,\ldots,s_0$ are single-row matrices of the form

$$H_u = \left[\frac{1}{(\beta_u-\alpha_1)^{b_u-1}} \quad \frac{1}{(\beta_u-\alpha_2)^{b_u-1}} \quad \cdots \quad \frac{1}{(\beta_u-\alpha_n)^{b_u-1}} \right],$$

the matrix H_I is a single row matrix of the form

$$H_I = \begin{bmatrix} 1 & \alpha & \cdots & \alpha^{m-1} & 0 \end{bmatrix},$$

where α is a primitive element of $\mathbb{F}_{2^m}$, and finally the matrix H_J is a row matrix of length $m+1$ of the form,

$$H_J = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \end{bmatrix}.$$

The code $\mathscr{C}_P$ has length $n = 2^m + ms_0$, redundancy $n - k \leq mt + s_0$, and distance $d \geq 2t + 1$ (Sugiyama et al., 1976). See Chapters 8 and 11.

BIBLIOGRAPHY

Berlekamp, E. (1968). *Algebraic coding theory*. New York: McGraw-Hill (cit. on pp. 6, 7).

— (1974). *Key papers in the development of coding theory*. New York: IEEE press (cit. on pp. 30, 61).

Berrou, C., A. Glavieux and P. Thitimajshima (1993). 'Near Shannon Limit Error-Correcting Coding: Turbo Codes'. In: *Proc. IEEE International Conference on Communications*. Geneva, Switzerland, pp. 1064–1070 (cit. on p. 7).

Bezzateev, S. and N. Shekhunova (2008). 'Chain of Separable Binary Goppa Codes and Their Minimal Distance'. In: *IEEE Transactions on Information Theory* 54.12, pp. 5773—5778 (cit. on pp. 123, 124, 129, 131).

Bezzateev, S. V. and N. A. Shekhunova (1995). 'Subclass of binary Goppa codes with minimal distance equal to the design distance'. In: *Information Theory, IEEE Transactions on* 41.2, pp. 554—555 (cit. on pp. 129, 131).

Blahut, R. E. (2008). *Algebraic Codes on Lines, Planes and Curves*. New York: Cambridge (cit. on pp. 44, 50, 61, 62).

Blake, I. et al. (1998). 'Algebraic-geometry codes'. In: *Information Theory, IEEE Transactions on* 44.6, pp. 2596 –2618 (cit. on pp. 49, 52, 54, 56).

Bose, R. C. and D. K. Ray Chaudhuri (Mar. 1960). 'On a class of error correcting binary group codes'. In: *Information and Control* 3.1, pp. 68–79 (cit. on pp. 6, 35).

Bosma, W., J. Cannon and C. Playoust (1997). 'The MAGMA algebra system I: The user language'. In: *J. Symbolic Comput.* 24 (3-4), pp. 235–265 (cit. on pp. 90, 93, 99, 100, 112, 130, 143, 150).

Brouwer, A. (1998). 'Bounds on the size of linear codes'. In: *Handbook of Coding theory: Volume I*. Ed. by V.S. Pless and W.C. Huffman. Amsterdam: Elsevier (cit. on pp. 20, 90).

Brouwer, A. E. and T. Verhoeff (Mar. 1993). 'An updated table of minimum-distance bounds for binary linear codes'. In: *Information Theory, IEEE Transactions on* 39.2, pp. 662 –677 (cit. on p. 90).

Cai, J. et al. (2005). 'An Efficient Solution to Packet Loss : Erasure Correcting Codes'. In: *Proc. Fourth IASTED International Conference (CSN'05)*, pp. 224 – 229 (cit. on pp. 61, 85).

Calabi, L. and E. Myrvaagnes (Oct. 1964). 'On the minimal weight of binary group codes (Corresp.)' In: *Information Theory, IEEE Transactions on* 10.4, pp. 385 – 387 (cit. on p. 90).

Canteaut, A. and F. Chabaud (1998). 'A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511'. In: *Information Theory, IEEE Transactions on* 44, pp. 367–378 (cit. on p. 27).

Cox, David A., John Little and Donald O'Shea (2007). *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra, 3/e (Undergraduate Texts in Mathematics)*. Secaucus: Springer-Verlag (cit. on p. 35).

Delsarte, P. (1975). 'On subfield subcodes of modified Reed-Solomon codes (Corresp.)' In: *Infosrmation Theory, IEEE Transactions on* 21.5, pp. 575 –576 (cit. on p. 24).

Ding, Cunsheng, H. Niederreiter and Chaoping Xing (2000). 'Some new codes from algebraic curves'. In: *Information Theory, IEEE Transactions on* 46.7, pp. 2638–2642 (cit. on pp. 95, 97).

Elias, P. (1955). 'Coding for noisy channels'. In: *IRE Conv. Rec.* 4 (cit. on pp. 5, 7).

Feng, G and T. N. Rao (1993). *Reflections on Decoding Algebraic-Geometric Codes up to the Designed Minimum Distance*. Electronic Article. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.47.1970&rep=rep1&type=pdf (cit. on pp. 7, 61).

Forney, G. D. and D. J. Costello (2007). 'Channel Coding: The Road to Channel Capacity'. In: *Proceedings of the IEEE* 95.6, pp. 1150–1177 (cit. on pp. 6, 30).

Forney G., Jr. (1970). 'Convolution Codes I: Algebraic Structure'. In: *Information Theory, IEEE Transactions on* 16, pp. 720–738 (cit. on p. 6).

Fossorier, M. P. C. and Shu Lin (Sept. 1995). 'Soft-decision decoding of linear block codes based on ordered statistics'. In: *Information Theory, IEEE Transactions on* 41.5, pp. 1379–1396 (cit. on pp. 61, 76).

Gallager, R. (1962). 'Low-Density Parity-Check Codes'. In: *IRE Trans. Inform. Theory* IT-8, pp. 21–28 (cit. on p. 7).

Garcia, A. and L. Quoos (2001). 'A Construction of Curves Over Finite Fields'. In: *ACTA Arithmetica* 98.2 (cit. on p. 101).

Garey, M. R. and D. S. Johnson (1979). *Computers and Intractability*. San Francisco: W.H. Freeman and Company (cit. on p. 131).

Geer, Gerard van der and Marcel van der Vlugt (2000). 'Kummer Covers with Many Points'. In: *Finite Fields and Their Applications* 6.4, pp. 327 –341 (cit. on p. 101).

Geer, Gerard van der et al. (2009). *Manypoints: A table of curves with many points*. Online available at http://www.manypoints.org (cit. on pp. 92, 100).

Golay, M. J. E. (1949). 'Notes on digital coding'. In: *Proc. IRE (corresp.)* 37 (cit. on p. 5).

Goppa, V. D. (1970). 'A New Class of Linear Error Correcting Codes'. In: *Probl. Peredachi Inf.* 6 (3) (cit. on pp. 6, 37, 41).

— (1971). 'A Rational Representation of Codes and (L,g)-Codes'. In: *Probl. Peredachi Inf.* 7 (3) (cit. on pp. 41, 105, 107, 173).

— (1972). 'Codes Constructed on the Base of (L, g)-Codes'. In: *Probl. Peredachi Inf.* 8 (2) (cit. on p. 105).

— (1988). *Geometry and Codes*. Dordrecht: Kluwer Academic Publishers (cit. on pp. 6, 8, 95).

Gorenstein, D. and N. Zierler (June 1961). 'A class of error correcting codes in p^m symbols'. In: *J. Soc. Ind. Appl. Math.* 9 (cit. on pp. 6, 35).

Grassl, M. (2006). 'Searching for linear codes with large minimum distance'. In: *Discovering Mathematics With MAGMA: Reducing Abstract to Concrete*. Ed. by Wieb Bosnan and John Cannon. Algorithms and Computations in Mathematics. Berlin: Springer-Verlag (cit. on p. 20).

— (2007). *Bounds on the minimum distance of linear codes and quantum codes*. Online available at `http://www.codetables.de`. Accessed on 21/12/2010 (cit. on pp. 48, 90, 91, 93, 95, 105, 112–116, 118–121, 123, 124, 128, 130, 145, 150, 167, 168, 171, 173).

— (2010). Private communication (cit. on p. 101).

Grassl, M. and G. White (2004). 'New good linear codes by special puncturings'. In: *Information Theory, 2004. ISIT 2004. Proceedings. International Symposium on*, p. 454 (cit. on pp. 93, 123–125).

Guruswami, V. and M. Sudan (1999). 'Improved decoding of Reed-Solomon and algebraic-geometry codes'. In: *Information Theory, IEEE Transactions on* 45.6, pp. 1757 –1767 (cit. on pp. 7, 43).

Hamming, R. W. (1950). 'Error detecting and error correcting codes'. In: *Bell Syst. Tech. J.* 29 (cit. on pp. 5, 30).

Helgert, H. and R. Stinaff (May 1973). 'Minimum-distance bounds for binary linear codes'. In: *Information Theory, IEEE Transactions on* 19.3, pp. 344 –356 (cit. on p. 90).

Hocquenhem, A. (1959). 'Codes correcteurs d'erreurs'. In: *Chiffres* 2 (cit. on pp. 6, 35).

Hoholdt, T. et al. (1998). *Handbook of coding theory*. Amsterdam: Elsevier (cit. on p. 56).

Johnston, M. and R. A. Carrasco (2005). 'Construction and performance of algebraic-geometric codes over AWGN and fading channels'. In: *Communications, IEE Proceedings-* 152.5, pp. 713–722 (cit. on p. 80).

Justesen, J. (1972). 'A class of constructive, asymptotically good algebraic codes'. In: *IEEE Trans. Inform. Theory* 18, pp. 652–656 (cit. on p. 6).

Justesen, J. et al. (1989). 'Construction and decoding of a class of algebraic geometry codes'. In: *Information Theory, IEEE Transactions on* 35.4, pp. 811–821 (cit. on pp. 56, 64).

Kasami, T., L. Costello and W. Peterson (1966). 'Some Results on the Weight Distribution of BCH Codes'. In: *Information Theory, IEEE Transactions on* 2.12, p. 274 (cit. on p. 156).

Koetter, R. and A. Vardy (2003). 'Algebraic soft-decision decoding of Reed-Solomon codes'. In: *Information Theory, IEEE Transactions on* 49.11, pp. 2809–2825 (cit. on p. 7).

Kohnert, Axel (2009). '(l,s)-extension of linear codes'. In: *Discrete Mathematics* 309.2, pp. 412 –417. URL: http://arxiv.org/pdf/cs.IT/0701112 (cit. on pp. 123, 124, 126, 127).

Lachaud, G. et al. (1995). 'Introduction to the Special issue on Algebraic Geometry Codes'. In: *Information Theory, IEEE Transactions on* 41.6, p. 1545 (cit. on p. 54).

Leonard, D. A. (1996). 'A generalized Forney formula for algebraic-geometric codes'. In: *Information Theory, IEEE Transactions on* 42.4, pp. 1263 –1268 (cit. on p. 72).

Leung, Ka Hin, San Ling and Chaoping Xing (2002). 'New binary linear codes from algebraic curves'. In: *Information Theory, IEEE Transactions on* 48.1, pp. 285 –287 (cit. on pp. 95, 102, 103).

Lidl, R. and H. Niederreiter (1986). *Introduction to finite fields and their applications*. Cambridge: Cambridge Univeristy Press (cit. on pp. 13, 15, 22, 139, 140, 156).

Lim, K. C. and Y. L. Guan (May 2006). 'Improved Code Shortening for Block and Product Codes'. In: *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*. Vol. 3, pp. 1367 –1371 (cit. on p. 124).

Liu, C. (1999). 'Determination of Error Values for Decoding Hermitian Codes with the Inverse Affine Fourier Transform'. In: *Fundamentals of Electronics, Communications and Computer Sciences, IEICE Transactions on* E82-A.10, pp. 2302–2305 (cit. on p. 72).

MacKay, D. J. C. and R. M. Neal (1996). 'Near Shannon Limit Performance of Low-Density Parity-Check Codes'. In: *IEEE Electron. Lett.* 32.18, pp. 1645–1646 (cit. on p. 7).

MacWilliams, F. J. and N. J. A. Sloane (1983). *The Theory of Error-Correcting Codes (North-Holland Mathematical Library)*. Amsterdam: North Holland (cit. on pp. 6, 9, 13, 19–22, 24, 27, 36, 37, 41–44, 74, 89, 93, 108, 110, 123, 124, 136, 155, 156, 159, 162, 164).

Massey, J. L. (1969). 'Shift register synthesis and BCH decoding'. In: *IEEE Trans. Inform. Theory* 15 (cit. on pp. 6, 7).

Massimo, Giullietti (2003). *Notes on Algebraic-Geometric Codes*. URL: http://www.math.kth.se/math/forskningsrapporter/Giulietti.pdf (cit. on p. 49).

Mattson, H. F. and G. Solomon (1961). 'A new treatment of error of Bose Chaudhari codes'. In: *J. Soc. Industr. Appl. Math.* 9 (cit. on p. 30).

Muller, D. E. (1954). 'Application of boolean algebra to switching circuit design and error detection'. In: *IRE Trans. Electron. Comput.* EC-3 (cit. on p. 5).

Niederreiter, H., C.P. Xing and K. Y. Lam (1999). 'A new construction of algebraic-geometry codes'. In: *Appl. Algebra Engrg. Comm. Comput.* 9.5 (cit. on p. 95).

Ozbudak, F. and H. Stichtenoth (1999). 'Constructing codes from algebraic curves'. In: *Information Theory, IEEE Transactions on* 45.7 (cit. on p. 95).

Patterson, N. (1975). 'The algebraic decoding of Goppa codes'. In: *Information Theory, IEEE Transactions on* 21.2, pp. 203 –207 (cit. on p. 6).

Peterson, W. and E. Weldon (1972). *Error-correcting Codes*. Cambridge: MIT Press (cit. on pp. 7, 82).

Peterson, W. W. (1960). 'Encoding and error-correction procedures for Bose-Chaudhuri codes'. In: *IRE Trans. Inform. Theory* 3 (cit. on p. 6).

Proakis John G.and Salehi, Masoud (2008). *Digital Communications*. New York: McGraw Hill (cit. on pp. 26, 82, 89).

Reed, I. S. (1954). 'A class of multiple error correcting codes and the decoding scheme'. In: *IRE Trans. Inform. Theory* IT-4 (cit. on p. 5).

Reed, I.S and G. Solomon (1960). 'Polynomial Codes over certain fields'. In: *J. Soc. Ind. Appl. Math.* 8 (cit. on pp. 6, 35).

Sakata, S. (1988). 'Finding a minimal set of linear recurring relations capable of generating a given finite two-dimensional array'. In: *J. Symb. Comput.* 5.3, pp. 321–327 (cit. on p. 62).

— (2010). *On the BMS algorithm*. Electronic article. Last accessed April, 2010. URL: http://www.ricam.oeaw.ac.at/specsem/srs/groeb/download/BMSalgWc.pdf (cit. on pp. 61, 62, 73).

Sakata, S. et al. (1995). 'Fast decoding of algebraic-geometric codes up to the designed minimum distance'. In: *Information Theory, IEEE Transactions on* 41.6, pp. 1672–1677 (cit. on pp. 7, 62, 65).

Schimd, W and R Shurer (2004). *Mint: A database for optimal net parameters*. Online available at http://mint.sbg.ac.at. Accessed on 2009-12-21 (cit. on pp. 48, 90, 95, 100, 103, 123, 176).

Shabat, V. (2001). 'Curves with many points'. PhD thesis. Amsterdam: Univ. of Amsterdam. URL: http://www.science.uva.nl/math/Research/Dissertations/Shabat2001.text.pdf (cit. on pp. 100, 101).

Shannon, C. E. (1948). 'A mathematical theory of communication'. In: *Bell System Technical Journal* 48 (cit. on pp. 3, 4).

Shokrollahi, M. A. and H. Wasserman (1999). 'List decoding of algebraic-geometric codes'. In: *Information Theory, IEEE Transactions on* 45.2, pp. 432–437 (cit. on p. 7).

Shu, Lin and Daniel J. Costello (2004). *Error Control Coding, Second Edition*. New Jersey: Prentice Hall (cit. on pp. 13, 44).

Simonis, J. (July 2000). 'Adding a parity-check bit'. In: *Information Theory, IEEE Transactions on* 46.4, pp. 1544 –1545 (cit. on p. 19).

Sloane, N., S. Reddy and Chin-Long Chen (July 1972). 'New binary codes'. In: *Information Theory, IEEE Transactions on* 18.4, pp. 503 –510 (cit. on pp. 20, 30, 115, 131, 150).

Sloane, N. J. A. (1972). 'A survey of constructive coding theory, and a table of binary codes of highest known rate'. In: *Discrete Mathematics* 3.1-3, pp. 265 –294 (cit. on p. 90).

Stern, J. (1989). 'A method for finding codewords of small weight'. In: *Proceedings of the 3rd International Colloquium on Coding Theory and Applications*. London, UK: Springer-Verlag, pp. 106–113 (cit. on p. 28).

Sudan, M. (2001). *Coding theory: Tutorial and Survey*. URL: http://people.csail.mit.edu/madhu/papers/2001/focs01-tut.pdf (cit. on p. 5).

Sugiyama, Y. et al. (1976). 'Further results on Goppa codes and their applications to constructing efficient binary codes'. In: *Information Theory, IEEE Transactions on* 22.5 (cit. on pp. 105, 106, 108, 109, 118, 173, 181, 182).

Tjhai, C. J. (2007). 'A Study of Linear Error Correcting Codes'. PhD thesis. Devon: Univ. of Plymouth (cit. on p. 75).

Tomlinson, M. et al. (2007). 'Analysis of the distribution of the number of erasures correctable by a binary linear code and the link to low-weight codewords'. In: *IET Communications* 1, pp. 539–548 (cit. on pp. 28, 29, 74).

Tomlinson, M. et al. (2011). 'New Binary Codes From Extended Goppa Codes'. Submitted to the 3rd International Castle Meeting for Coding Theory and Applications (cit. on p. 116).

Tsfasman, M. A., S. G. Vladut and T. Zink (1982). 'On Goppa codes which are better than the Varshamov-Gilbert bound'. In: *Math. Nacr.* 109, pp. 21–28 (cit. on pp. 7, 45, 95).

Tzeng, K. and K. Zimmermann (1975). 'On extending Goppa codes to cyclic codes (Corresp.)' In: *Information Theory, IEEE Transactions on* 21.6 (cit. on p. 106).

Van-Lint, J. H. (1990). 'Algebraic geometry codes'. In: *Coding theory and design theory: Part I: Coding Theory*. Ed. by D. Ray-Chaudhari. New York: Springer-Verlag, p. 137 (cit. on p. 49).

Vardy, A. (Nov. 1997). 'The intractability of computing the minimum distance of a code'. In: *Information Theory, IEEE Transactions on* 43.6, pp. 1757 –1766 (cit. on pp. 26, 130).

Verhoeff, T. (1987). 'An updated table of minimum-distance bounds for binary linear codes'. In: *Information Theory, IEEE Transactions on* 33.5, pp. 665 –680 (cit. on p. 90).

Véron, P. (2005). 'Proof of Conjectures on the True Dimension of Some Binary Goppa Codes'. In: *Des. Codes Cryptography* 36 (3), pp. 317–325 (cit. on p. 129).

Walker, Judy L. (2000). *Codes and Curves*. Rhode Island: American Mathematical Society (cit. on pp. 46, 49, 96).

Wolf, J. K. (1969). 'Adding two information symbols to certain non-binary BCH codes and some applications'. In: *Bell Syst. Tech. J.* 48 (cit. on pp. 135, 138, 156, 157, 159, 162, 169).

Xing, C., H. Niederreiter and K. Y. Lam (1999a). 'A generalization of algebraic-geometry codes'. In: *Information Theory,IEEE Transactions on* 45.7, pp. 2498–2501 (cit. on pp. 7, 93, 95, 97, 98, 102).

Xing, C. P., H. Niederreiter and K. Y. Lam (1999b). 'Constructions of algebraic-geometry codes'. In: *IEEE Trans. Inform. Theory* 45.4, pp. 1186 –1193 (cit. on p. 95).

Xing, Chaoping and Sze Ling Yeo (2007). 'New linear codes and algebraic function fields over finite fields'. In: *Information Theory, IEEE Transactions on* 53.12, pp. 4822–4825 (cit. on pp. 93, 95, 97).

Zierler, G. (1960). 'On decoding linear error-correcting codes'. In: *Information Theory, IRE Transactions on* 6 (cit. on p. 30).

ACRONYMS

AG algebraic geometry. 6–8, 45, 85, 175

AWGN additive white Gaussian noise. 4, 8, 10, 11, 26, 61, 175

BCH Bose Chaudhari Hocquenghem. 6, 10, 35

BEC binary erasure channel. 26

BMA Berlekamp Massey algorithm. 8, 10, 61, 175

BMSA Berlekamp Massey Sakata algorithm. 8, 10, 61, 62, 85, 175

BSC binary symmetric channel. 26

CPU central processing unit. 30

DMC discrete memoryless channel. 25

GRS generalised Reed Solomon. 36, 43

LDPC low density parity check. 7, 8, 177

MDS maximum distance separable. 36

RAM random access memory. 30

RS Reed Solomon. 6, 10, 35, 61

INDEX