Subspace Codes and
Distributed Storage Codes

Netanel Raviv
Subspace Codes and Distributed Storage Codes

Research Thesis

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Netanel Raviv

Submitted to the Senate of the Technion — Israel Institute of Technology
Av 5776 Haifa August 2016
The research thesis was done under the supervision of Prof. Tuvi Etzion in the Computer Science Department.

The author would like to express his sincere gratitude to Prof. Tuvi Etzion for his endless support. The author would also like to thank his collaborators for their valuable contributions (in alphabetical order): Prof. Eli Ben-Sasson, Dr. Ariel Gabizon, Prof. Frank Kschischang, Prof. Muriel Médard, Prof. Ronny Roth, Dr. Natalia Silberstein, Dr. Antonia Wachter-Zeh, Prof. Eitan Yaakobi.

The generous financial support of the Technion, the Israeli Science Foundation (10/12), the IBM Ph.D. fellowship, the Mitacs foundation, and the Aharon and Ephraim Katzir grant is gratefully acknowledged.
Contents

Abstract 1
Abbreviations and Notations 2
A Note for the Reader 3

1 Introduction 7

2 Research Methods 12

I Journal Papers 15

3 Equidistant Codes in the Grassmannian 16

3.1 Introduction 16
3.2 Trivial Codes and Trivial Constructions 20

3.2.1 Partial spreads 21
3.2.2 Extension of a code 22
3.2.3 Sunflowers 22
3.2.4 Optimal \((k - 1)\)-intersecting equidistant codes in \(G_q(n, k)\) 23
3.2.5 Orthogonal subspaces 24

3.3 Constructions of Large Non-Sunflower Equidistant Codes 25

3.3.1 Construction using the Plücker embedding 25
3.3.2 Do larger equidistant codes exist? 29
3.3.3 Large non-sunflower equidistant codes 30
3.3.4 Equidistant codes in \(P_q(n)\) 30

3.4 Equidistant Rank Metric Codes 31

3.5 Recursive Construction of Equidistant Subspace Codes 34
3.6 Conclusion and Problems for Future Research .............. 36

4 Subspace Polynomials and Cyclic Subspace Codes .......... 41
  4.1 Introduction ........................................ 41
  4.2 Subspaces and their Subspace Polynomials ................. 44
  4.3 Cyclic Subspace Codes ................................ 50
     4.3.1 Codes with Full Length Orbits .................... 51
     4.3.2 Codes with degenerate orbits ..................... 54
  4.4 Conclusions and Future Work ............................ 57

5 Some Gabidulin Codes Cannot Be List Decoded Efficiently at any Radius .......... 67
  5.1 Introduction ......................................... 68
  5.2 Preliminaries ......................................... 72
  5.3 Sets of Subspaces Polynomials with Mutual Top Coefficients ...................... 73
  5.4 Improved Bounds on List Decodability of Gabidulin Codes ...................... 77
  5.5 Bounds for Constant-Dimension Subspace Codes .......................... 85
  5.6 Conclusions and Future Work .................................. 89
  5.A Correction ........................................... 97

II Conference Papers .............................................. 100

6 Distributed Storage Systems Based on Intersecting Subspace Codes .......... 101
  6.A Conference Version .................................... 101
     6.A.1 Introduction ..................................... 101
     6.A.2 Preliminaries .................................... 105
     6.A.3 Repair ............................................ 108
        6.A.3.1 Minimum Bandwidth Repair .................. 108
        6.A.3.2 Local Repair ................................ 109
        6.A.3.3 Parallel Repair .............................. 109
     6.A.4 Reconstruction and Modification ....................... 110
        6.A.4.1 Reconstruction ................................ 110
        6.A.4.2 Modification ................................ 111
     6.A.5 Assignment of Vectors .................................. 111
     6.A.6 Conclusions ....................................... 113
### 7.B.5 Construction of an MSR Code with Three Parities
- **7.B.5.1 Three Parities from One Matching**
- **7.B.5.2 Three Parities from Two Matchings**
- **7.B.5.3 Construction of Matchings for Three Parities**

### 7.B.6 An Improved Construction over a Larger Field
- **7.B.6.1 A Code from One Matching**
- **7.B.6.2 A Code from Two Matchings**

### 8 Coding for Locality in Reconstructing Permutations

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.A</td>
<td>Conference Version</td>
<td>221</td>
</tr>
<tr>
<td>8.A.1</td>
<td>Introduction</td>
<td>222</td>
</tr>
<tr>
<td>8.A.2</td>
<td>Previous Work</td>
<td>224</td>
</tr>
<tr>
<td>8.A.3</td>
<td>Bounds</td>
<td>225</td>
</tr>
<tr>
<td>8.A.3.1</td>
<td>Upper Bounds</td>
<td>226</td>
</tr>
<tr>
<td>8.A.3.2</td>
<td>Lower Bound</td>
<td>226</td>
</tr>
<tr>
<td>8.A.4</td>
<td>High Rate Constructions</td>
<td>227</td>
</tr>
<tr>
<td>8.A.4.1</td>
<td>Concatenation of Short Permutations</td>
<td>227</td>
</tr>
<tr>
<td>8.A.4.2</td>
<td>Concatenation of Range-Restricted Permutations</td>
<td>228</td>
</tr>
<tr>
<td>8.A.4.3</td>
<td>Extended Construction from Error-Correcting Codes</td>
<td>229</td>
</tr>
<tr>
<td>8.A.4.4</td>
<td>High-Locality Construction From Multi-Permutations</td>
<td>232</td>
</tr>
<tr>
<td>8.A.5</td>
<td>Additional Results</td>
<td>234</td>
</tr>
<tr>
<td>8.A.6</td>
<td>Discussion and Open Problems</td>
<td>235</td>
</tr>
</tbody>
</table>

### 8.B Unpublished Full Version

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.B.1</td>
<td>Introduction</td>
<td>239</td>
</tr>
<tr>
<td>8.B.1.1</td>
<td>Motivation</td>
<td>242</td>
</tr>
<tr>
<td>8.B.2</td>
<td>Previous Work</td>
<td>245</td>
</tr>
<tr>
<td>8.B.3</td>
<td>Bounds</td>
<td>248</td>
</tr>
<tr>
<td>8.B.3.1</td>
<td>Upper Bounds</td>
<td>248</td>
</tr>
<tr>
<td>8.B.3.2</td>
<td>Lower Bound</td>
<td>251</td>
</tr>
<tr>
<td>8.B.4</td>
<td>High Rate Constructions</td>
<td>256</td>
</tr>
<tr>
<td>8.B.4.1</td>
<td>The Alternating Group</td>
<td>256</td>
</tr>
<tr>
<td>8.B.4.2</td>
<td>Concatenation of Short Permutations</td>
<td>258</td>
</tr>
<tr>
<td>8.B.4.3</td>
<td>Concatenation of Range-Restricted Permutations</td>
<td>260</td>
</tr>
<tr>
<td>8.B.4.4</td>
<td>Extended Construction from Error-Correcting Codes</td>
<td>261</td>
</tr>
<tr>
<td>8.B.4.5</td>
<td>High-Locality Construction From Multi-Permutations</td>
<td>265</td>
</tr>
<tr>
<td>8.B.5</td>
<td>Construction of Specific Families</td>
<td>268</td>
</tr>
<tr>
<td>8.B.5.1</td>
<td>Light Permutations by the Infinity Norm</td>
<td>268</td>
</tr>
<tr>
<td>8.B.5.2</td>
<td>MinMax Permutations</td>
<td>270</td>
</tr>
</tbody>
</table>
### 8.B.6 Efficient Representation of Permutations
- **8.B.6.1 Light Permutations by the Infinity Norm - An Alternative Representation**
- **8.B.6.2 Supporting Arbitrary Powers - a framework**
- **8.B.6.3 Beating LRCs - a proof of concept**

### 8.B.7 Discussion and Open Problems

### III Unpublished Papers

### 9 Asymptotically Optimal Regenerating Codes Over Any Field

#### 9.A ArXiV Version
- **9.A.1 Introduction**
- **9.A.2 Previous Work**
- **9.A.3 Preliminaries**
  - **9.A.3.1 Companion matrices and representation of extension fields**
  - **9.A.3.2 Kronecker products and cyclotomic cosets**
- **9.A.4 Nearly MBR codes**
  - **9.A.4.1 Construction**
  - **9.A.4.2 The proximity of NMBR codes to MBR codes**
- **9.A.5 Nearly MSR codes**
  - **9.A.5.1 Construction**
  - **9.A.5.2 The proximity of NMSR codes to MSR codes**
- **9.A.6 Connection to subspace codes**
  - **9.A.6.1 Construction of an every-$d$ independent set in $G_q(b+c, \frac{b+c}{d})$**
  - **9.A.6.2 Nearly MBR codes from subspace codes**
  - **9.A.6.3 Vandermonde matrix structure**
- **9.A.7 Discussion and future research**

### Discussion
## List of Figures

5.1 An illustration of the proof of Theorem 25. The proof of Theorem 26 is similar. The ball around $c_R$ of radius $\tau$ contains the words $c_{R-P_i}$ for $P_i \in \mathcal{P}$, where $|\mathcal{P}| = \left\lfloor \frac{n/g}{(n-\tau)/g} \right\rfloor q^{\ell \cdot 79}$

6.1 Subspace framework for the reconstruction of the file $x$ from nodes $1, \ldots, k$.

6.2 Subspace framework for the repair of node $d+1$ from nodes $1, \ldots, d$.

6.3 The structure of a storage node.

7.1 A perfect colored matching on the complete (2-uniform hyper)graph with four vertices. The edges of the matching appear in bold, and the nodes are colored in $\heartsuit, \diamondsuit$ such that there is no monochromatic edge.

7.2 Two matchings in the complete (2-uniform hyper)graph which satisfy the pairing condition. Each edge in one is monochromatic in the other, and vice versa.

8.1 The equivalence of a transversal in a cyclic Latin square, an arrangement of non attacking semi-queens in a torodial chess-board, and a permutation $\sigma$ such that $\text{Id} + \sigma$ is a permutation.

8.2 The dependency graph between the symbols of a MinMax permutation $\pi \in \mathcal{M}_L$. Nodes represent subsets of symbol locations, where $\{i\}$ is denoted by $i$. An edge $(i, S)$ indicates that the repair of symbol $\pi_i$ requires exactly one arbitrary symbol $\pi_j, j \in S$.
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>The resulting radius in Lemma 30: If $G_{n,k}$ cannot be list decoded efficiently for the radius $\tau$, then the punctured code $G_{n-s,k}$, $s &lt; n - k + 1$, cannot be list decoded efficiently for this radius as well. The rightmost column provides $\tau$ as a function of $\tau'$ and $s$, where the unique decoding radius of $G_{n-s,k}$ is $\tau' - 1$. The given values for $\tau'$ are simple calculations which follow from $n - k - s$ being either even or odd.</td>
<td>85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1</td>
<td>Comparison of our code with some previously known MSR codes with two parities.</td>
<td>141</td>
</tr>
<tr>
<td>7.2</td>
<td>Comparison of our code with some previously known MSR codes with three parities.</td>
<td>141</td>
</tr>
<tr>
<td>7.3</td>
<td>Comparison of our code with some previously known MSR codes with two parities.</td>
<td>158</td>
</tr>
<tr>
<td>7.4</td>
<td>Comparison of our code with some previously known MSR codes with three parities.</td>
<td>159</td>
</tr>
<tr>
<td>7.5</td>
<td>Eigenspaces of $A_Z, A_Z', A_Z'',$ and $A_Z^*$.</td>
<td>204</td>
</tr>
<tr>
<td>7.6</td>
<td>Eigenspaces of $A_Z^2, A_Z'^2, A_Z''^2,$ and $A_Z^*^2$.</td>
<td>206</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>Summary of the results in Section 8.B.4.</td>
<td>257</td>
</tr>
<tr>
<td>8.2</td>
<td>Sizes of certain permutations sets for small values of $n$.</td>
<td>280</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>A comparison of parameters between our NMBR codes (Subsection 9.A.4.1) and the PM-MBR codes [24, Sec. IV] for general $n, k, d$.</td>
<td>304</td>
</tr>
</tbody>
</table>
9.2 A comparison of parameters between our NMBR codes (Subsection 9.A.4.1) and the PM-MBR codes [24, Sec. IV] for several common parameters $n, k, d$. . . . . . . . . . . . . . . . 305

9.3 A comparison of parameters between our MSR codes (Subsection 9.A.5.1) and the PM-MSR codes [24, Sec. V] for general $n, k, d = 2k - 2$. . . . . . . . . . . . . . . . . . . . . . . . . . . 312

9.4 A comparison of parameters between our code (Subsection 9.A.5.1) and the PM-MSR code [24, Sec. V] for several common parameters $n, k, d = 2k - 2$. . . . . . . . . . . . . . . . . . . . . . 313
Abstract

The interest in subspace codes has increased lately due to their application in error correction for random network coding. A subspace code is a collection of subspaces of a vector space under the subspace distance $d_s(U,V) = \dim U + \dim V - 2 \dim (U \cap V)$. In this dissertation we discuss both purely mathematical and practical questions which involve subspace codes.

The theoretical part of this work concerns a few special cases of subspace codes. Equidistant subspace codes, in which the distance between any two codewords is identical, are discussed, and bounds and constructions are given. Cyclic subspace codes, which are closed under multiplication by a field element are discussed, and the first non-trivial construction of those is given by using subspace polynomials. The latter topic is related to showing the limits of list-decoding Gabidulin and subspace codes, improving previously known bounds.

The practical aspect of this work concerns distributed storage systems. A distributed storage system is comprised of storage nodes with communication links, and the system is required to maintain reliability over time. The aforementioned equidistant subspace codes are shown to be useful for devising a minimum bandwidth regenerating code (MBR) for distributed storage systems, and a similar approach is shown to attain asymptotically optimal regenerating codes over any field. In addition, a minimum storage regenerating code (MSR) is constructed using subspace techniques, and finally, the question of local erasure correction in permutations is discussed, under the motivation of performing updates to a distributed storage system.
Abbreviations and Notations

$q$ — A power of prime.
$\mathbb{F}_q$ — A finite field with $q$ elements.
$\mathbb{F}^n_q$ — A vector space of dimension $n$ over $\mathbb{F}_q$.
$\mathbb{F}_{q^n}$ — A finite field with $q^n$ elements, seen as an extension of degree $n$ of $\mathbb{F}_q$.
$\mathcal{G}_q(n,k)$ — The Grassmannian, the set of all $k$-dimensional subspaces of $\mathbb{F}^n_q$. 
A Note for the Reader

Publications

This dissertation is based on the following publications.


Notice that papers [2], [4], [5], [7], and [8] were published (will be published) in the IEEE International Symposium on Information Theory (ISIT) 2015 (2016). In addition, the following were presented in the meetings of Action IC1104 of the European Cooperation in Science and Technology (COST). These meeting have no published proceedings.

Paper [1] was presented in the COST meeting joint with the 4th International Castle Meeting on Coding Theory and Applications (4ICMCTA), Palmela, Portugal, September 2014.

Paper [3] was presented in Algebra, Codes and Networks Bordeaux, France, June 2014.


Following the instructions of the Technion’s Graduate School, I list herein my contribution to the aforementioned works, which were all done in collaboration with other researchers. Multiple publications of the same research will not be included more than once.

Paper [1] After a discussion between Dr. Ariel Gabizon and myself about the open problem of explicit construction of cyclic subspace codes, Dr. Gabizon presented the problem to Prof. Ben-Sasson, which suggested the use of subspace polynomials, and in particular, the gap property of those. After a short research I was able to find an explicit polynomial which solves the conjecture of existence of cyclic subspace codes of distance $2k - 2$, and after further research, I have found another polynomial which enlarges this construction using the Frobenius automorphism. Both of these constructions were enabled by a lemma I proved regarding the size of the orbit with respect to a given subspace
polynomial. A further study by Dr. Gabizon and myself yielded the results about cyclic subspace codes with degenerate orbits, which led me to discover [6].

Paper [3] Most of the work (bounds, simple constructions) was done together with Prof. Etzion. My main contribution to this work was finding the central construction of an equidistant subspace codes using the Plücker embedding, including the recursive construction which begins with the Fano plane, and the corresponding equidistant rank metric code, which yielded [5].

Paper [4] The topic of efficient updates for distributed storage systems was suggested by Prof. Médard during my visit to MIT. After a brief study of the topic I suggested to use sets of permutations with locality, and devised several simple constructions and bounds. After establishing a collaboration with Prof. Yaakobi, I was able to find, with his tremendous help and support, several new constructions (e.g., with multi-permutations) and bounds.

Paper [5] The connection between equidistant subspace codes and distributed storage was found in a paper by Hollmann while writing the research proposal of this dissertation. After this discovery, the application of the equidistant subspace code from [3] was soon to be discovered by me, and developed in tight collaboration with Prof. Etzion.

Paper [6] The improvement to the list-decoding bound for Gabidulin codes was found by me. The further implication to subspace codes was found by Dr. Wachter-Zeh, which also contributed deeply to understanding the exact connections to previous bounds. As a result of a talk in our faculty coding theory seminar, Prof. Ronny Roth observed that the applicable rate could be reduced from $\frac{1}{2}$ to $\frac{1}{3}$, which was later reduced even further by Dr. Wachter-Zeh and myself to $\frac{1}{5}$.

Paper [8] Dr. Silberstein introduced me to this problem and provided a small example of an $(\mathcal{A}, S)$-set with $4 \times 4$ matrices which she found by a computer search. I was able, with the help of Prof. Etzion and Dr. Silberstein, to enhance this small example to a construction which improves the state-of-the-art codes for distributed storage.
Note that Paper [8] was recently submitted to the IEEE transactions on information theory, and Paper [4] was recently submitted to Designs, Codes, and Cryptography. At the time of writing these lines, no reply was given yet from the committees of both journals.

Finally, Chapter 9 provides the most recent addition to this dissertation. In the summer of 2016 I was interning under Prof. Frank Kschischang in the University of Toronto, Canada, with the generous support of the Mitacs foundation under the Globalink Canada-Israel Innovation Initiative. Given Prof. Kschischang’s interest in subspace codes, and in coordination with Prof. Tuvi Etzion, in was suggested to attempt at improving [5] during this internship. A preliminary version of the obtained results is given in Chapter 9, and hopefully, a online version will be available by the time the interested reader will read these lines.

Structural Road-map

This dissertation contains three parts. The first is dedicated to the full journal versions of the aforementioned papers, the second one is dedicated to the conference versions, and the third is dedicated to unpublished works. Each topic in given in a separate chapter within these parts. Since the conference versions are rather succinct, each conference paper is followed by the unpublished full version. Note that as a result, certain parts of the research are given multiple times to maintain the flow for the convenience of the reader.

Bibliography

In accordance with the instruction of the Technion’s Graduate School, each paper contains its own list of references. The separate bibliography which is given in p. 337 lists the references which are given in the Introduction, the Discussion, and the Research methods chapters.
Chapter 1

Introduction

Coding theory is a connecting bridge between computer science and mathematics. Crossing this bridge stands as a prominent obstacle in engineering, which undermines considerable efforts of both theorists and practitioners. The research which is presented in this dissertation aims to strengthen the foundations on which this bridge is build, by studying subspace codes, distributed storage codes, and the connection of the two.

Modern day interest in subspace codes stems mainly from their application to network coding, a topic which gained increasing interest in the past decade due to its ability to improve network information flow. In a network of servers which are connected by communication links of limited capacity, it was shown [1, 16, 18] that the admissible transmission rate is bounded from above by the capacity of the minimum cut in the underlying graph. This optimal rate can be attained by allowing the servers to perform linear operations between the packets of information they receive, where each packet is modeled as a vector over a finite field. This line of research, which concerns encoding, decoding, and code constructions, is called network coding.

Beyond all aspects of efficiency lies the aspect of reliability, which may be seen as the cornerstone of coding theory. Physical limitations of hardware may cause errors in transmitted packets. Due to the propagative nature of networks, a single erroneous packet might render an entire transmission unreadable. This inherent flaw has put an ominous question mark on the applicability of network coding, placing tremendous scientific efforts in peril. Fortunately, the work of [17] came to the rescue, showing that subspace codes are precisely what is needed for error correction in network coding.
A subspace code is a collection of subspaces of a vector space over a finite field, under the so-called *subspace distance*, defined as $d_S(U, V) = \dim U + \dim V - 2 \dim(U \cap V)$, where $U, V$ are two codewords (that is, subspaces). These mathematical objects were studied in the past under different motivations (e.g., [30]), and this intriguing application has ignited the interest in a wide variety of applicative and theoretical questions about them. In the sequel, we discuss some of these questions that were addressed in our research.

In parallel with the prosperity of network coding, a new challenge which involves coding theory has emerged. Modern Internet companies, such as Facebook and Dropbox, have started to report an exponential growth in the volumes of their stored data. These tremendous amount of data, which contain highly sensitive and personal information, must be stored reliably over time. For this purpose, Distributed Storage Systems (DSSs) are employed, which consist of storage nodes connected by communication links. As in the case of network coding, the stored data might be lost due to hardware failures, and once again coding theory is called up to provide a solution by adding redundant information to the system. However, when applying classic *erasure codes*, objects that were devised in the 1960’s for communication scenarios, unexpected issues arise. When employing classic erasure codes, the amount of network traffic which is required in order to repair a failed node is equivalent to the entire stored dataset, a fact which renders them unusable in practice.

A network coding flavored analysis has lead to the introduction of *re-generating codes* [6], as a means to overcome high repair bandwidth by downloading only a fraction of the content from each remaining disk. A different approach, in which each node is repairable by downloading the entire content of a small number of adjacent disks, was enabled by recent advances in *Locally Recoverable Codes* (LRCs) [31]. The latter part of this dissertation addresses these two approaches, that were shown to directly involve subspaces and subspace codes by [15, 23, 32].

The research which is presented in this dissertation began with a purely mathematical study of *equidistant subspace codes* [9]. A code is called equidistant if any two subspaces in it intersect mutually in a subspace of the same dimension. As under other topics in coding theory, the two basic challenges are to bound the maximal possible size of these codes, and to
construct codes which attain it. This topic was previously studied for ordinary codes, and yet for subspace codes very little research was done, and only a few small constructions were known. In our research we devised a bound on the maximal size of such codes, using parallel results on ordinary codes, and provided a novel construction, by using the Plücker embedding. Julius Plücker was a 19-th century mathematician and physicist, whose contribution to the field of algebraic geometry, and hence to our research, remained rather esoteric among coding theorists. Using his techniques, we managed to provide a construction of a constant dimension equidistant subspace codes, which remains one of the largest nontrivial construction to date. Some of the problems that were mentioned as possible future research were since addressed by others [2, 13].

Albeit being purely mathematical, it was soon to be discovered that our construction of an equidistant subspace code [9] has a direct application in the construction of good storage codes [23]. Although the resulting system is equivalent in performance to a system which employs regenerating codes (and in particular, Minimum Bandwidth Regenerating (MBR) codes), and even though the connection between subspace codes and distributed storage was known [5] Example 3.2, our construction of a subspace-based system is the first one of its kind to present comparable performance to state-of-the-art regenerating codes. This connection is yet to be studied, and has the potential of tackling the many challenges which still remain in the construction of codes for storage.

The following problem which was addressed, once again, a mathematical one, is the construction of cyclic subspace codes [3]. A subspace code is called cyclic if any subspace in it is a shift of another. Cyclic subspace codes are of particular interest, since several small examples of optimal subspace codes, found by a computer search, turned out to be cyclic [5]. Albeit several studies on this topic [11, 33], no general construction was known. In our work, using the well-known subspace polynomials, we obtained the first explicit construction of cyclic subspace codes of non-trivial distance, and thus confirming a conjecture by [10, 11, 33].

While studying subspace polynomials [9], we have discovered that these objects are used to show limits of list-decoding of Reed-Solomon [4], rank-metric [35], and subspace codes [27]. A list-decoding problem is a close variant of the classic decoding problem, in which the erroneous data need
not to be decoded uniquely, but rather to a small list of possible options. Given that this list is small enough, list-decoding enables to tolerate a larger number of errors in the data. Since our work revealed several facts regarding the coefficient structure of subspace polynomials, we were able to revise and improve the results of [4, 27, 35]. In particular, we were able to prove that certain Gabidulin codes, which are the basis of the aforementioned subspace code construction by [17], cannot be listdecoded efficiently at all. This resolved an open problem by [7, 14, 35].

Motivated by the connection we have found between subspaces and storage [23], we were intrigued to find out that the construction of Minimum Storage Regenerating (MSR) codes involves a sophisticated structure of subspace and matrices, which we called an \((A, S)\)-set [32]. This set consists of matrices \(A_i\) which form the generator matrix of the code, and subspace \(S_i\) which are used for the repair process. To enable the construction of an optimal storage system, the \((A, S)\)-set must satisfy a certain list of requirements called the subspace condition. In our work, we have found that such an \((A, S)\)-set can be constructed using graph-theoretic tools, namely, colored perfect matchings in a complete hypergraph. This technique enabled the construction of optimal storage systems, with two or three redundancy nodes. As oppose to previous constructions, our construction drastically reduces the size of the underlying finite field, and thus bringing theoretical concepts into the realm of feasibility.

Further study of storage systems revealed certain questions about updates to the stored data. That is, given a system which currently stores a dataset, how can we perform small modifications to this dataset, without the need to encode the whole system anew? A previous work on this topic [28] has not addressed one natural type of updates, which is permutations (such as in a cut-and-paste operation). A brief study has led us to a conclusion that in very simple storage systems, such as RAID-4, it is impossible to perform certain small permutation changes in the file without incurring major updates in storage nodes. In our work [26] we deviate from previous approaches [28], and suggest to store the permutation separately.

Separate storage of permutations raises the question of local erasure correction for permutations, a question which was studied before only for ordinary strings (e.g., under the title of LRCs [31]). To enable efficient storage of a permutations, one may either focus on sets of permutations which
present \textit{locality}, that is, any symbol of the permutation can be computed from a small number of other symbols, or devise storage techniques which make use of the inherent combinatorial structure of permutations. In this work, we proved several upper bounds on the maximal possible size of a set of permutations with locality, and provided several simple construction which attain it for low values of locality. In cases where the bound is not attained we constructed such sets using a variety of tools, such as permutation polynomials, Reed-Solomon codes, multi-permutations, and known techniques from the field of data structures \cite{19}. Finally, we discovered that \textit{optimal} linear storage of permutations, that is, one which enables local erasure correction by addition, is intimately related to a variant of the famous non-attacking queens problem in chess. Albeit seeing a significant progress recently \cite{8}, this problem is yet to be solved.

The most recent addition to this dissertation, which is given in the last part, was motivated by the aforementioned application of equidistant subspace codes to distributed storage systems. In this work, a powerful class of regenerating codes called \textit{product matrix codes} \cite{22} was employed in order to obtain asymptotically optimal codes over any field. A central tool in this construction is an algebraic structure which we called an \textit{every-$k$ independent} set of subspaces. This structure is a set of subspaces such that any $k$ of them span the entire vector space, and follows easily from a representation of a Vandermonde matrix over an extension field as a block matrix using conventional notions from field theory. This connection to field extension representations led us to discover that subspace codes are not necessary in this context, and the results may follow also from more classical concepts, most notably, Kronecker products and cyclotomic cosets. This application of a classic purely mathematical concept to distributed storage clearly serves, together with equidistant subspace codes, $(A,S)$-sets, and cyclic subspace codes, as part of the few building blocks that this dissertation contributes to the connecting bridge between mathematics and computer science.
Chapter 2

Research Methods

This chapter provides an overview of the mathematical techniques which were employed in the research which led to this dissertation, with particular emphasis on methods which were developed during this work.

It is not uncommon in coding theory, and in science in general, that an esoteric mathematical technique, which was developed purely as a mind game of eager theoreticians, was later found to be applicable in a seemingly unrelated area of study. Some of the most prominent examples of such scenarios are the application of Boolean algebra for construction of computer hardware, and the application of Riemannian geometry to Einstein’s general theory of relativity. A much more modest instance of this scenario occurred at the outset of this research [9, 23]. Julius Plücker’s method of embedding subspaces in the projective place, nicknamed “the Plücker embedding”, was found to be applicable in construction of equidistant subspace codes, and later on, in the construction of distributed storage systems.

Briefly speaking, Plücker embedding is a function which maps an element from $G_q(n,k)$ (the set of subspaces of dimension $k$ of a vector space of dimension $n$ over the field $\mathbb{F}_q$ with $q$ elements) to the projective place $\mathbb{P}_q^{(n-1)}$. The remarkable fact in this embedding is that it is defined over a spanning matrix of the subspace rather than the subspace itself, and yet, it is well-defined even though the spanning matrix is not unique.

Shortly after the research of subspace codes as a means for error correction was initiated, Plücker’s technique was employed to obtain results about subspace code, mainly under the motivation of list-decoding (e.g., [34]).
However, it was used as a means of expressing a ball in the subspace metric, rather than for construction. The use of subspace codes for distributed storage systems was briefly discussed in the past, but only exponentially smaller codes were known [15].

The central tool in the next line of works [3, 25] is linearized polynomials, which were introduced in the early 1930’s [20], and are a well-established notion in coding theory and mathematics. A polynomial \( L(x) \in \mathbb{F}_q[x] \) is called a linearized polynomial if 
\[
L(x) = \alpha_0 + \alpha_1 \cdot x^q + \ldots + \alpha_k \cdot x^{q^k}
\]
for some field elements \( \alpha_0, \ldots, \alpha_k \in \mathbb{F}_q \) and some integer \( k \). A monic linearized polynomial is called a \textit{subspace polynomial} with respect to \( \mathbb{F}_q \) if it divides \( x^{q^n} - x \).

In the novel work of Koetter and Kschischang [17], linearized polynomial are the main tool in constructing their near-optimal subspace code. The main novelty of our work is to consider the subspace polynomials as representatives of the subspace in the code itself, rather than as a means in the construction. This interpretation has revealed that subspace codes with certain subspace polynomial structure possess several desired properties, such as being cyclic. Finally, applying sets of subspace polynomials to obtain bounds on list-decodability of Reed-Solomon and Gabidulin codes is certainly not a novelty of this work [4, 35]. Yet, a construction of a certain cyclic subspace code we have found yielded a considerable improvement of the known bound for list-decoding Gabidulin codes.

The results of [24] are also based on an existing technique. As mentioned in the introduction, the construction of Minimum Storage Regenerating codes for distributed storage systems is based on an algebraic structure called an \((A, S)\)-set. This structure was thoroughly studied in the past, and yet, only constructions with (sometimes considerably) larger field were known. Our methods of interpreting the \((A, S)\)-set in graph theoretic tools enabled, using some technical work, to significantly reduce the field size. In a nutshell, \((A, S)\)-sets consists of pairs of a subspace and a matrix such that in each pair, the subspace is an \textit{independent} subspace of the matrix, and in distinct pairs, each subspace is an \textit{invariant} subspace of the matrix. Interpreting the subspace as being spanned by vectors which constitute nodes in a perfect matching in a complete graph, enabled us to perform a finer analysis than the one performed in [32]. This analysis has led to simultaneous diagonalizability of matrices, which enabled a reduction in the field size.

The question of locality in permutations [26] was raised under the general
motivation of performing efficient updates in a distributed storage system. The methods in this chapter are deeply connected to various objects in coding theory and combinatorics. Among these objects are Reed-Solomon codes, permutation polynomials, multi-permutations, disjoint cycle representation, and most notably, the famous non-attacking queens problem.

Finally, the preliminary unpublished results which are given in Chapter 9 were also obtained through subspace codes. The original motivation of the research which led to these results was improving and generalizing [23], in which equidistant subspace codes were used to construct distributed storage systems. Our results were obtained by interleaving techniques from a powerful class of distributed storage codes called product matrix codes [22], and from an algebraic object which we call an every-$k$ independent set of subspaces. Albeit being of independent mathematical interest, it was soon to be discovered that every-$k$ independent sets are not necessary; our results also follow using standard notions from field theory and from matrix analysis, i.e., extension fields and Kronecker products (a common tool for solving matrix equations). Nevertheless, the connection to subspace codes is given in its entirety for the sake of a complete description of our line of research, and of course, for the sake of mathematical curiosity.
Part I

Journal Papers
Chapter 3

Equidistant Codes in the Grassmannian

Tuvi Etzion and Netanel Raviv

Abstract

Equidistant codes over vector spaces are considered. For $k$-dimensional subspaces over a large vector space the largest code is always a sunflower. We present several simple constructions for such codes which might produce the largest non-sunflower codes. A novel construction, based on the Plücker embedding, for 1-intersecting codes of $k$-dimensional subspaces over $\mathbb{F}_q^n$, $n \geq \binom{k+1}{2}$, where the code size is $\frac{q^{k+1}-1}{q-1}$ is presented. Finally, we present a related construction which generates equidistant constant rank codes with matrices of size $n \times \binom{n}{2}$ over $\mathbb{F}_q$, rank $n-1$, and rank distance $n-1$.

3.1 Introduction

Equidistant codes in the Hamming scheme are the bridge between coding theory and extremal combinatorics. A code is called equidistant if the distance between any two distinct codeword is equal to a given parameter $d$. A code which is represented as collection of subsets will be called $t$-intersecting if the intersection between any two codewords is exactly $t$. A binary code of length $n$ in the Hamming scheme is a collection of binary words of length
A binary code of length \( n \) with \( m \) codewords can be represented as a binary \( m \times n \) matrix whose rows are the codewords of the code. The Hamming distance between two codewords is the number of positions in which they differ. The weight of a word in the number of nonzero entries in the word. A code \( C \) is called a constant weight code if all its codewords have the same weight. The minimum Hamming distance of the code is the smallest distance between two distinct codewords. An \((n,d,w)\) code is a constant weight code of length \( n \), weight \( w \), and minimum Hamming distance \( d \). Optimal \( t \)-intersecting equidistant codes (which must be also constant weight) have two very interesting families of codes, codes obtained from projective planes and codes obtained from Hadamard matrices [5].

Recently, there have been lot of new interest in codes whose codewords are vector subspaces of a given vector space over \( \mathbb{F}_q \), where \( \mathbb{F}_q \) is the finite field with \( q \) elements. The interest in these codes is a consequence of their application in random network coding [23]. These codes are related to what are known as \( q \)-analogs. The well known concept of \( q \)-analogs replaces subsets by subspaces of a vector space over a finite field and their sizes by the dimensions of the related subspaces. A binary codeword can be represented by a subset whose elements are the nonzero positions. In this respect, constant dimension codes are the \( q \)-analog of constant weight codes. For a positive integer \( n \), the set of all subspaces of \( \mathbb{F}_q^n \) is called the projective space \( \mathcal{P}_q(n) \). The set of all \( k \)-dimensional subspaces (\( k \)-subspaces in short) of \( \mathbb{F}_q^n \) is called a Grassmannian and is denoted by \( \mathcal{G}_q(n,k) \). The size of \( \mathcal{G}_q(n,k) \) is given by the \( q \)-binomial coefficient \( \binom{n}{k}_q \), i.e.

\[
|\mathcal{G}_q(n,k)| = \binom{n}{k}_q \triangleq \frac{(q^n-1)(q^{n-1}-1)\cdots(q^{n-k+1}-1)}{(q^k-1)(q^{k-1}-1)\cdots(q-1)}.
\]

Similarly, the set of all \( k \)-subspaces of a subspace \( V \) is denoted by \( \binom{V}{k}_q \). It turns out that the natural measure of distance in \( \mathcal{P}_q(n) \) is given by

\[
d_S(X,Y) \triangleq \dim X + \dim Y - 2\dim(X \cap Y),
\]

for all \( X,Y \in \mathcal{P}_q(n) \). This measure of distance is called the subspace distance and it is the \( q \)-analog of the Hamming distance in the Hamming space. Finally, an \([n,d,k]_q\) code is a subset of \( \mathcal{G}_q(n,k) \) (constant dimension code), having minimum subspace distance \( d \), where the minimum subspace distance
of the code is the smallest subspace distance between any two codewords. In the sequel, when distance will be mentioned it will be understood from the context if the Hamming distance or the subspace distance is used.

The main goal of this paper is to consider the $t$-intersecting constant dimension codes (which are clearly equidistant). In this context interesting constructions of such codes were given in [17, 18], but their size is rather small.

Some optimal binary equidistant codes form a known structure from extremal combinatorics called a partial projective plane. This structure was defined and studied by [15]. An important concept in this context is the sunflower. A binary constant weight code of weight $w$ is called a sunflower if any two codewords intersect in the same $t$ coordinates. A sunflower $S \subseteq G_q(n,k)$ is a $t$-intersecting equidistant code in which any two codewords $X, Y \in S$ intersect in the same $t$-subspace $Z$. The $t$-subspace $Z$ is called the center of $S$ and is denoted by $\text{Cen}(S)$.

An upper bound on the size of $t$-intersecting binary constant weight code with weight $w$ was given in [5, 6]. If the size of such code is greater than $(w - t)^2 + (w - t) + 1$ then the code is a sunflower. This bound is attained when $t = 1$, $w = q + 1$, where $q$ is a prime power and the codewords are the characteristic vectors of length $\left[ \frac{n}{1} \right]_q$ which represent the lines of the projective plane of order $q$. Except for some specific cases [16] no better bound is known. This bound can be adapted for $t$-intersecting constant dimension codes with dimension $k$ over $\mathbb{F}_q$. We can view the characteristic binary vectors with weight $\frac{q^k - 1}{q - 1}$ as the codewords, which implies intersection of size $\frac{q^k - 1}{q - 1}$ between codewords and hence we have the following bound.

**Theorem 1** If a $t$-intersecting constant dimension code of dimension $k$ has more than \( \left( \frac{q^k - q^t}{q - 1} \right)^2 + \frac{q^k - q^t}{q - 1} + 1 \) codewords then the code is a sunflower.

The bound of Theorem 1 is rather weak compared to the known lower bounds. The following conjecture is attributed to Deza.

**Conjecture 1** If a $t$-intersecting code in $G_q(n,k)$ has more than $\left[ \frac{k+1}{1} \right]_q$ codewords then the code is a sunflower.

\(^1\text{See } \text{http://www.math.ucla.edu/ chowdhury/research/shadow-int-vec-space-extended.pdf}\)
The term $t$-intersecting code is different from the highly related term $t$-intersecting family \cite{9, 12}. A family of $k$-subsets of a set $X$ is called $t$-intersecting if any two $k$-subsets in the family intersect in at least $t$ elements \cite{9}. The celebrated Erdős-Ko-Rado theorem determines the maximum size of such family. The $q$-analog problem was considered for subspaces in \cite{19, 12}. For the $q$-analog problem, a $t$-intersecting family is is a set of $k$-subspaces whose pairwise intersection is at least $t$. In \cite{12} it was proved that

\begin{equation}
\frac{\binom{n-t}{k-t} q^k + \binom{2k-t}{k} q^k}{\binom{n}{k} q^k}.
\end{equation}

Although Theorem 2 concerns fundamentally different families from the ones discussed in this paper, some intriguing resemblance is evident. One can easily see that there exist two simple families of subspaces attaining the bound of Theorem 2. The first is the set of all $k$-subspaces sharing at least a fixed $t$-subspace. The second is the set of all $k$-subspaces contained in some $(2k-t)$-subspace These elements are closely related to the terms sunflower and ball discussed in details in Section 3.2.

One concept which is heavily connected to constant dimension codes is rank-metric codes. For two $k \times \ell$ matrices $A$ and $B$ over $\mathbb{F}_q$ the rank distance is defined by

\begin{equation}
d_R(A, B) \triangleq \text{rank}(A - B).
\end{equation}

A $[k \times \ell, q, \delta]$ rank-metric code $C$ is a linear code, whose codewords are $k \times \ell$ matrices over $\mathbb{F}_q$; they form a linear subspace with dimension $q$ of $\mathbb{F}_q^{k \times \ell}$, and for each two distinct codewords $A$ and $B$ we have that $d_R(A, B) \geq \delta$.

There is a large literature on rank-metric code and also on the connections between rank-metric codes and constant dimension codes, e.g. \cite{3, 10, 13, 14, 27, 28}. Given a rank-metric code, one can form from it a constant dimension code, by lifting its matrices \cite{10, 28}. Optimal constant dimension codes can be derived from optimal codes of a subclass of rank-metric codes, namely constant rank codes. In a constant rank code all matrices have the same rank. The connection between optimal codes in this class and optimal codes...
constant dimension codes was given in [14]. This also can motivate a re-
search on equidistant constant rank codes which we will also discuss in this
paper.

The rest of this paper is organized as follows. In Section 3.2 we consider
trivial codes and trivial constructions. The trivial equidistant constant di-
mension codes are the $q$-analogs of the trivial binary equidistant constant
weight codes. It will be very clear that the trivial $q$-analogs are not so
trivial. In fact for most parameters we don’t know the size of the largest
trivial codes. We will also define two operations which are important in
constructions of codes. The first one is the extension of a code (or the ex-
tended code of a given code) preserves the triviality of the code. The second
operation is the orthogonality (or the orthogonal code of a given code) pre-
serves triviality only in one important set of parameters. In Section 3.3 we
present our main result of the paper, a construction of 1-intersecting code
in $G_q(n, k)$, $n \geq \binom{k+1}{2}$, whose size is $\binom{k+1}{k+1}_q$. The construction is based on
the Plücker embedding [2]. We will show one case in which a larger code
by one is obtained, which falsify Conjecture [1] in this case. We also con-
sider the largest non-sunflower $t$-intersecting codes in $G_q(n, k)$ based on our
discussion. Finally, we consider the size of the largest equidistant code in
all the projective space $P_q(n)$. In Section 3.4 we use the Plücker embed-
ding to form a constant rank code over $F_q$, with matrices of size $n \times \binom{n}{2}$,
rank $n - 1$, rank distance $n - 1$, and $q^n - 1$ codewords. The technique of
Section 3.4 is used in Section 3.5 for a recursive construction of exactly the
same codes constructed in Section 3.3. Conclusion and open problems for
future research are given in Section 3.6.

### 3.2 Trivial Codes and Trivial Constructions

In this section we will consider first trivial codes. A binary constant weight
equidistant code $C$ of size $m$ is called trivial is every column of $C$ has $m$ or
$m - 1$ equal entries. Now, we define a $q$-analog for a constant dimension
equidistant code. A constant dimension equidistant code $C \subset G_q(n, k)$ will
be called trivial if one of the following two conditions holds.

1. Each element $x$ of $F_q^n$ is either contained in all $k$-subspaces of $C$, no
   $k$-subspace of $C$, or exactly one $k$-subspace of $C$.  


2. If $T$ is the smallest subspace of $\mathbb{F}_q^n$ which contains all the $k$-subspaces of $C$ then $T$ is a $(k + 1)$-subspace.

A trivial code which satisfies the first condition is a sunflower. A trivial code which satisfies the second condition will be called a ball. We will examine now the types of constant dimension equidistant codes which satisfy one of these two conditions. We will discover that in some cases triviality does not mean that it is easy to find the optimal (largest) trivial code.

3.2.1 Partial spreads

Two subspaces $X, Y \in G_q(n, k)$ are called disjoint if their intersection is the null space, i.e. $X \cap Y = \{0\}$. A partial $k$-spread (or a partial spread in short) in $G_q(n, k)$ is a set of disjoint subspaces from $G_q(n, k)$. If $k$ divides $n$ and the partial spread has $\frac{q^n - 1}{q^k - 1}$ subspaces then the partial spread is called a $k$-spread (or a spread in short). A partial spread in $G_q(n, k)$ is the $q$-analog of a 0-intersecting constant weight code of length $n$ and weight $k$. The number of $k$-subspaces in the largest partial spread of $G_q(n, k)$ will be denoted by $E_q[n, k]$. The known upper and lower bounds on $E_q[n, k]$ are summarized in the following theorems. The first three well-known theorems can be found in [11].

**Theorem 3** If $k$ divides $n$ then $E_q[n, k] = \frac{q^n - 1}{q^k - 1}$.

**Theorem 4** $E_q[n, k] \leq \left\lfloor \frac{q^n - 1}{q^k - 1} \right\rfloor - 1$ if $n \not\equiv 0 \pmod{k}$.

**Theorem 5** Let $n \equiv r \pmod{k}$. Then, for all $q$, we have

$$E_q[n, k] \geq \frac{q^n - q^k(q^r - 1) - 1}{q^k - 1}$$

The next theorem was proved in [21] for $q = 2$ and for any other $q$ in [1].

**Theorem 6** If $n \equiv 1 \pmod{k}$ then $E_q[n, k] = \frac{q^n - q}{q^k - 1} - q + 1 = \sum_{i=1}^{n+1} q^{ik+1} + 1$.

Theorem 6 was extended for the case where $q = 2$ and $k = 3$ in [8] as follows.
Theorem 7 If \( n \equiv c \pmod{3} \) then \( E_2[n, 3] = \frac{2^n - 2^c}{2} - c \).

The upper bound implied by Theorem 6 was improved for some cases in [7].

Theorem 8 If \( n = k\ell + c \) with \( 0 < c < k \), then \( E_q[n, k] \leq \sum_{i=0}^{\ell-1} q^{i\ell} + c - \Omega - 1 \), where \( 2\Omega = \sqrt{1 + 4q^k(q^k - q^c)} - (2q^k - 2q^c + 1) \).

### 3.2.2 Extension of a code

An \((n, d, w)\) constant weight equidistant code \( C \) is extended to an \((n+1, d, w+1)\) constant weight equidistant code \( E(C) \) by adding a column of ones to the code. Similarly, an \([n, d, k]_q\) constant dimension equidistant code \( C \) is extended to an \([n+1, d, k+1]_q\) constant dimension equidistant code \( E(C) \) as follows. We first define \( \mathbb{F}_q^{n+1} \) by \( \mathbb{F}_q^{n+1} \triangleq \{(x, \alpha) \mid x \in \mathbb{F}_q^n, \alpha \in \mathbb{F}_q\} \). For a subspace \( X \in \mathcal{G}_q(n, k) \) let \((X, 0)\) be a subspace in \( \mathcal{G}_q(n+1, k) \) defined by \((X, 0) \triangleq \{(x, 0) \mid x \in X\} \). Let \( v \in \mathbb{F}_q^{n+1} \setminus \{(x, 0) \mid x \in \mathbb{F}_q^n\} \) and \( C \subset \mathcal{G}_q(n, k) \). We define the extended code \( E(C) \) by

\[
E(C) \triangleq \{(X, 0) \cup \{v\} \mid X \in C\}.
\]

The following theorem can be easily verified.

**Theorem 9** If \( C \) and \( C \) are trivial codes then the extended codes \( E(C) \) and \( E(C) \) are also trivial codes.

We note that the extended code \( E(C) \) of a trivial code \( C \) is not unique, but all such extended codes are isomorphic. An \([n, d, k]_q\) constant dimension equidistant code \( C \) can be extended \( \ell \) times to an \([n+\ell, d, k+\ell]_q\) constant dimension equidistant code. This extended code will be denoted by \( E^\ell(C) \).

### 3.2.3 Sunflowers

A partial spread is clearly a 0-intersecting sunflower. For a given \( n, k \) such that \( 0 < k < n \) we construct the largest \( t \)-intersecting sunflower in \( \mathcal{G}_q(n, k) \) by using the following two simple theorems.

**Theorem 10** If \( S \) is a \( t \)-intersecting sunflower in \( \mathcal{G}_q(n, k) \) then \( E(S) \) is a \((t+1)\)-intersecting sunflower in \( \mathcal{G}_q(n+1, k+1) \).
Theorem 11 Let S be a t-intersecting sunflower in Gq (n, k) and let X be an
(n−t)-subspace of Fnq such that X ⊕Cen(S) = Fnq . Then the set {X ∩Y | Y ∈
S} is a partial (k − t)-spread in X.
If S is the largest partial (k − t)-spread in Gq (n − t, k − t) then by Theorem 10 we have that E t (S) is a t-intersecting sunflower in Gq (n, k). By
Theorem 11 we have that E t (S) is the largest t-intersecting sunflower in
Gq (n, k).

3.2.4

Optimal (k−1)-intersecting equidistant codes in Gq (n, k)

In this subsection we present a construction for optimal (k − 1)-intersecting
equidistant code in Gq (n, k) for any n ≥ k + 1.
Theorem 12 An optimal non-sunflower (k−1)-intersecting equidistant code
 
in Gq (n, k), n ≥ k + 1 has k+1
k q subspaces.
 
Proof. Let V be any (k + 1)-subspace of Fnq and let C = Vk q , i.e. C
consists of all k-subspaces of a (k + 1)- subspace. It is readily verified that
every two such k-subspaces intersect at a (k − 1)-subspace, and the size of
 
C is k+1
k q.
Let C0 be an equidistant (k − 1)-intersecting code in Gq (n, k). Let
X ∈ C0 , and let S1 , S2 be any two distinct sunflowers in C0 such that
Cen(S1 ), Cen(S2 ) ⊆ X. It is easy to verify that for every X1 ∈ S1 , X2 ∈ S2
(which implies that dim(X1 ∩ X2 ) = k − 1) we have that |X1 ∩ X2 ∩ X| =
q k−2 − 1 which implies that |X1 ∩ X2 ∩ X c | = (q k−1 − 1) − (q k−2 − 1) =
(q − 1)q k−2 .
We prove now that for every X1 ∈ S1 and every Y, Z ∈ S2 the sets
A , X1 ∩Y ∩X c and B , X1 ∩Z ∩X c are mutually disjoint. Since Y, Z ∈ S2
and Cen(S2 ) ⊆ X, it follows that Y and Z do not intersect outside X, i.e.
(Y ∩ Z) ∩ X c = ∅. If A ∩ B =
6 ∅ or equivalently X1 ∩ Y ∩ Z ∩ X c 6= ∅ then
Y ∩ Z ∩ X c 6= ∅, a contradiction.
Therefore, in the set {X1 ∩ Y ∩ X c | Y ∈ S2 } there can be at most
(q k − q k−1 )/(q − 1)q k−2 = q disjoint subsets of size (q − 1)q k−2 . Hence, the
size of any sunflower other than S1 is at most q + 1. However, all of the
above arguments are applicable for any initial sunflower S1 . Therefore, each
sunflower whose center is inside X have at most q + 1 codewords, including
23

Technion - Computer Science Department - Ph.D. Thesis PHD-2017-01 - 2017


$X$ itself. $X$ may have at most $\binom{k}{k-1} q$ sunflower centres inside it, which yields that:

$$|C'| \leq 1 + q \cdot \binom{k}{k-1} = \binom{k+1}{k} = |C|.$$ 

Thus, the claim in the theorem follows.

**Corollary 1** An optimal non-sunflower equidistant $(k-1)$-intersecting code in $\mathcal{G}_q(n,k)$, $n \geq k+1$ consists of all $k$-subspaces of any given $(k+1)$-subspace. This code is a trivial code which satisfies the second condition, i.e., it is a ball.

### 3.2.5 Orthogonal subspaces

A simple operation which preserves the equidistant property of a code and its triviality in the Hamming scheme is the complement. Two binary words $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are complements if for each $i$, $1 \leq i \leq n$, we have $x_i + y_i = 1$, i.e., $x_i = 0$ if and only if $y_i = 1$. We say that $y$ is the complement of $x$ and denote it by $y = \bar{x}$. For a binary code $C$, the complement of $C$, $\bar{C}$ is defined by $\bar{C} = \{x \mid \bar{x} \in C\}$. It is easily verified that if a constant weight code $C$ is equidistant then also its complement is constant weight code and equidistant and if $C$ is trivial then also its complement $\bar{C}$ is trivial.

What is the $q$-analog operation for the complement? This question was discussed in details before [4]. We will use the orthogonality as the the $q$-analog for complement. For a constant dimension code $C \subseteq \mathcal{G}_q(n,k)$ we define the orthogonal code by $C^\perp \triangleq \{X \mid X^\perp \in C\}$. It is known [11, 23, 29] that for any two subspaces $X, Y \in \mathbb{F}_q^n$ we have $d_S(X,Y) = d_S(X^\perp, Y^\perp)$. This immediately implies the following result.

**Theorem 13** If $C$ is a $t$-intersecting equidistant code in $\mathcal{G}_q(n,k)$ then $C^\perp$ is a $t'$-intersecting equidistant code in $\mathcal{G}_q(n,n-k)$, where $t' = n - 2k + t$.

**Proof.** Clearly, $C^\perp$ is a code in $\mathcal{G}_q(n,n-k)$. Furthermore, $C$ and $C^\perp$ have the same minimum subspace distance $d$ which is also the distance between any two codeword of $C$ and between any two codewords of $C^\perp$. Therefore, $d = 2(k-t) = 2(n-k-t')$ which implies $t' = n - 2k + t$. ■

**Corollary 2** If $C$ a 0-intersecting code in $\mathcal{G}_q(n,k)$ (a partial spread) then $C^\perp$ is an $(n-2k)$-intersecting code in $\mathcal{G}_q(n,n-k)$.
Corollary 3 The smallest possible $n$ for which a $t$-intersecting code exists in $G_q(n, k)$ is $n = 2k - t$. The size of the largest such code is the size of the largest partial spread in $G_q(2k - t, k - t)$.

The next question we would like to answer is whether the orthogonal code of a trivial code is also a trivial code. The answer is that usually, the orthogonal code of a trivial code is not a trivial code. The only exception is given in the following theorem whose proof is easy to verify.

Theorem 14 $C$ is an optimal $(k-1)$-intersecting equidistant code in $G_q(n, k)$ (a ball) if and only if $C^\perp$ is a $(n-k-1)$-intersecting sunflower in $G_q(n, n-k)$.

3.3 Constructions of Large Non-Sunflower Equidistant Codes

In this section we will consider the size of the largest non-sunflower equidistant code in $G_q(n, k)$ and in $P_q(n)$. The main result which will be presented in subsection 3.3.1 is a construction of 1-intersecting codes in $G_q(n, k)$, $n \geq \binom{n+1}{2}$, whose size is $\left\lfloor \frac{k+1}{2} \right\rfloor q$. This construction will be based on the Plücker embedding. In subsection 3.3.2 we will show the only example we have found (by computer search) for which this construction is not optimal. In subsection 3.3.3 we will consider the largest non-sunflower $t$-intersecting codes in $G_q(n, k)$ based on our discussion. In subsection 3.3.4 we will consider the size of the largest equidistant code in the whole the projective space $P_q(n)$.

3.3.1 Construction using the Plücker embedding

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$, let $e_i$ denote the unit vector (of a given length) with an one in the $i$th coordinate, and for any $0 < k \leq n$ let $n_k \triangleq \binom{n}{k}$. We denote by $\mathbb{P}^{k-1}_q$ the set of projective points of $\mathbb{P}^k_q$. The set $\mathbb{P}^{k-1}_q$ is commonly referred to as the set $\mathbb{P}^k_q \sim$ where $\sim$ is an equivalence relation over $\mathbb{F}_q^k \setminus \{0\}$ defined as

$$x \sim y \iff \exists \lambda \in \mathbb{F}_q \setminus \{0\}, x = \lambda y.$$ 

It is widely known [2 25 26] that $G_q(n, k)$ can be embedded in $\mathbb{P}^n_k \setminus 1$ using the Plücker embedding, denoted by $P$. 

25
Given $U \in \mathcal{G}_q(n, k)$, let $M(U) \in \mathbb{F}^{k \times n}_q$ be some $k \times n$ matrix over $\mathbb{F}_q$ whose row span is $U$. Given any $k$-subset $\{i_1, \ldots, i_k\}$ of $[n]$, let $M(U)(i_1, \ldots, i_k)$ be the $k \times k$ sub-matrix of $M(U)$ consisting of columns $i_1, \ldots, i_k$. Consider the $\binom{n}{k}$ coordinates of $\mathbb{F}_q^n$ as numbered by $k$-subsets of $[n]$, and for each $k$-subset $\{i_1, \ldots, i_k\}$ of $[n]$ assume w.l.o.g that $i_1 < \ldots < i_k$. The function $P$ maps a subspace $U \in \mathcal{G}_q(n, k)$ to the equivalence class of the vector $v(U) \in \mathbb{F}_q^n$, defined as:

$$(v(U))_{\{i_1, \ldots, i_k\}} \triangleq \det M(U)(i_1, \ldots, i_k).$$

Namely, the coordinate $\{i_1, \ldots, i_k\}$ of the vector $v(U)$ is the determinant of the sub-matrix of $M(U)$ consisting of columns $i_1 < \ldots < i_k$ of $M(U)$. Formally, the function $P$ is defined as $P(U) = [v(U)]$, where $[v(U)]$ is the equivalence class of the vector $v(U)$ under the equivalence relation ~ defined earlier in this section. It is worth mentioning that the function $P$ is well-defined, as any choice of a matrix $M(U)$ whose row span is $U$ will result in the same equivalence class in $\mathbb{F}_q^{nk-1}$ [2]. In this paper, in order to maintain consistency with coding theory terminology, we identify the set $P_{\ell-1}$ by the set of all 1-subspaces of $\mathbb{F}_q^\ell$, namely, $\mathcal{G}_q(\ell, 1)$.

Another concept we use in this section is Steiner systems. A Steiner system $S(t, k, n)$ is a pair $(Q, B)$ where $Q$ is an $n$-set of elements (called points) and $B$ is a collection of $k$-subsets of $Q$ (called blocks), such that every $t$-subset of $Q$ is contained in exactly one block of $B$. A Steiner system can be described by its incidence matrix. This is a matrix $A = (a_{ij}), i \in [n], j \in [b], b = |S(t, k, n)|$, where if $Q = \{q_1, \ldots, q_n\}$ and $B = \{B_1, \ldots, B_b\}$ we have:

$$a_{ij} = \begin{cases} 1 & \text{if } q_i \in B_j \\ 0 & \text{if } q_i \notin B_j \end{cases}.$$ 

The following lemma, which is a key in the construction which follows, can be easily verified.

**Lemma 1** The rows of the matrix $A$ defined by a Steiner system $S(2, k, n)$ form an 1-intersecting equidistance code. The weight of each codeword in this code is $\frac{n-1}{k-1}$.

In the sequel we use the Plücker embedding, together with Lemma 1 to construct equidistant constant dimension codes.
**Theorem 15** For every integer \( n \geq 3 \) there exists an 1-intersecting equidistant code in \( \mathcal{G}_q(n_2, n-1) \) of size \( \binom{n}{1}_q \).

The main idea of the construction in the proof of Theorem 15 is to consider a set \( S_{q,n} \) of blocks, whose blocks are the elements of \( \mathcal{G}_q(n, 2) \) and its set of points is \( \mathcal{G}_q(n, 1) \) where a point \( X \in \mathcal{G}_q(n, 1) \) is incident with a block \( Y \in \mathcal{G}_q(n, 2) \) if \( X \subseteq Y \). \( S_{q,n} \) forms a Steiner system \( S(2, q+1, q(n-1) \cdot q-1) \) since every 2-subspace contains \( q+1 \) points and every pair of distinct 1-subspaces is contained in a unique 2-subspace. We embed \( \mathcal{G}_q(n, 2) \) into \( \mathbb{P}^{n_2-1}_q \) using the Plücker embedding, and show that given \( V \in \mathcal{G}_q(n, 1) \), the set \( PV = \bigcup_{U \in \mathcal{G}_q(n, 2), V \subseteq U} P(U) \) is the union of the Plücker embeddings of subspaces in \( \mathcal{G}_q(n, 2) \) which intersect at \( V \) and constitutes a vector space in \( \mathcal{G}_q(n_2, n-1) \). Therefore, we may use Lemma 1 to get an 1-intersecting equidistant code in \( \mathcal{G}_q(n_2, n-1) \) of size \( \binom{n}{1}_q = q(n-1) \). The proof of Theorem 15 relies on the following two lemmas.

**Lemma 2** \( |PV| = q^{n-1}. \)

**Proof.** Note that the number of 2-subspaces that contain a given 1-subspace is \( \frac{q^n-1}{q-1} \). Since each two distinct 1-subspaces \( P(U_1), P(U_2) \) intersect trivially, it follows that \( |PV| = \frac{q^{n-1}}{q-1} \cdot (q-1) + 1 = q^{n-1} \).

**Lemma 3** If \( V \in \mathcal{G}_q(n, 1) \) then \( PV \in \mathcal{G}_q(n_2, n-1) \).

**Proof.** Let \( v \) be an arbitrary nonzero vector in \( V \). Let \( r \) be an arbitrary index such that \( v_r \neq 0 \) and let \( B_v = \{ z^i \}_{i=1}^{n-1} \) be the set of \( n-1 \) distinct unit vectors of length \( n \) such that \( e_r \notin B_v \). If \( Z_i \triangleq \langle v, z^i \rangle, 1 \leq i \leq n-1 \), then by the definition of Plücker embedding, the projective point \( P(Z_i) \), considered as an 1-subspace of \( \mathbb{F}_q^{n_2} \), is the span of the vector \( p^i \) of length \( n_2 \) defined by

\[
p^i_{(s,t)} \triangleq \det_{(s,t)} \begin{pmatrix} v \\ z^i \\ z_s^i \\ z_t^i \\ \end{pmatrix} = v_s z_t^i - v_t z_s^i.
\] (3.1)

Recall that the coordinates of \( \mathbb{F}_q^{n_2} \) are identified with 2-subsets of \( [n] \) as mentioned earlier in this section.
We will now prove that $\langle \{p^i\}_{i=1}^{n-1}\rangle = P_V$, and since $|P_V| = q^{n-1}$ by Lemma 2, it implies that $P_V$ is an $(n-1)$-subspace of $\mathbb{F}_q^{n2}$.

Let $x \triangleq \sum_{i \in [n-1]} a_i p^i$ where $a_i \in \mathbb{F}_q$ for all $i \in [n-1]$. We will show that $x \in P_V$, i.e. that there exists a $\tilde{Z} \in \mathcal{G}_q(n_2, 2)$ such that $V \subseteq \tilde{Z}$ and $x \in P(\tilde{Z})$. As a consequence we have $\langle \{p^i\}_{i=1}^{n-1}\rangle \subseteq P_V$. Let

$$\tilde{z} \triangleq \sum_{i \in [n-1]} a_i z^i.$$  \hfill (3.2)

Clearly, $V \subseteq \langle v, \tilde{z} \rangle$, and since $\tilde{z}_r = 0, v_r \neq 0$ it follows that $\langle v, \tilde{z} \rangle$ is a 2-subspace of $\mathbb{F}_q^n$. By the definition of the Plücker embedding, $P(\langle v, \tilde{z} \rangle)$ is the span of the following vector $\tilde{x}$:

$$\tilde{x}_{\{s,t\}} \triangleq \det_{\langle s,t \rangle} \left( \frac{v}{\tilde{z}} \right)$$

$$= v_s \tilde{z}_t - v_t \tilde{z}_s$$

$$= v_s \left( \sum_{i \in [n-1]} a_i z^i \right) - v_t \left( \sum_{i \in [n-1]} a_i z^i \right)$$

by (3.2)

$$= \sum_{i \in [n-1]} a_i \left( v_s z^i - v_t z^i \right)$$

$$= \sum_{i \in [n-1]} a_i p^i_{\{s,t\}}$$

by (3.1)

$$= x_{\{s,t\}}$$

by the definition of $x$

Hence $\tilde{x} = x$ and therefore $\langle \{p^i\}_{i=1}^{n-1}\rangle \subseteq P_V$.

Now, let $x \in P_V$ and let $U_x \in \mathcal{G}_q(n, 2)$ be the unique 2-subspace such that $V \subseteq U_x$ and $x \in P(U_x)$. Hence, $U_x = \langle v, z \rangle$ for some $z \in \mathbb{F}_q^n$. By the definition of $\mathcal{B}_v$ every vector whose $r$th entry is 0 is in $\langle \mathcal{B}_v \rangle$. If $z_r \neq 0$ then we define

$$z' = v - \left( \frac{v_r}{z_r} \right) \cdot z \in U_x.$$ 

Clearly, $z'_r = 0$ and hence $U_x = \langle v, z' \rangle$. Thus, w.l.o.g we assume $z_r = 0$ and we may write $z = \sum_{i \in [n-1]} a_i z^i$ for some $a_i \in \mathbb{F}_q, i \in [n - 1]$. Since $U_x = \langle v, z \rangle$ it follows that the 1-subspace $P(U_x)$ is spanned by the vector
\((\det(s,t) \binom{u}{v})_{s,t})\subseteq [n]\). Hence, there exists some \(\lambda \in \mathbb{F}_q\) such that:

\[
x_{\{s,t\}} = \lambda \cdot \det_{\{s,t\}} \binom{v}{z} = \lambda (v_s z_t - v_t z_s)
\]

\[
= \lambda \left(v_s \cdot \sum_{i \in [n-1]} a_i z_t^i - v_t \cdot \sum_{i \in [n-1]} a_i z_s^i\right) \quad \text{since } z = \sum_{i \in [n-1]} a_i z^i
\]

\[
= \lambda \sum_{i \in [n-1]} a_i (v_s z_t^i - v_t z_s^i)
\]

\[
= \lambda \sum_{i \in [n-1]} a_i P_{\{s,t\}}^i \quad \text{by } (3.1)
\]

Therefore \(x \in \langle \{p^i\}_{i=1}^{n-1} \rangle\) which implies that \(P_V \subseteq \langle \{p^i\}_{i=1}^{n-1} \rangle\).

Thus we have proved that \(P_V = \langle \{p^i\}_{i=1}^{n-1} \rangle\), and as a consequence we have that \(P_V \in \mathcal{G}_q(n_2, n-1)\). \(\blacksquare\)

**Proof.** (of Theorem 15) Let \(C \subseteq \mathcal{G}_q(n_2, n-1)\) be the code defined by

\[
C \triangleq \{P_V \mid V \in \mathcal{G}_q(n, 1)\}.
\]

By Lemma 3 for each \(V \in \mathcal{G}_q(n, 1)\) we have that \(P_V\) is an \((n-1)\)-subspace of \(\mathbb{F}_q^{n_2}\) and hence \(C\) is well-defined. By Lemma 1 and the discussion on \(S_{q,n}\), it follows that \(C\) is an 1-intersecting equidistant code. \(\blacksquare\)

**Remark 1** After the paper was written, we found that Lemma 3 can also be obtained as a consequence of [20, Theorem 24.2.9, p.113] which discuss the theory of finite projective geometries. But, the proof of this Theorem requires more detailed theory which precedes it, while our proof is much shorter, simpler, and direct.

### 3.3.2 Do larger equidistant codes exist?

It is believed (Conjecture 1) that the largest non-sunflower 1-intersecting code in \(\mathcal{G}_q(n, k)\) has size at most \(\left\lfloor \frac{k+1}{2} \right\rfloor\). The following example consists of an 1-intersecting code \(C \subseteq \mathcal{G}_2(6, 3)\) of size 16, while \(\left\lfloor \frac{6}{1} \right\rfloor = 15\). \(C\) was found by a computer search.

Let \(\alpha\) be a primitive root of \(x^6 + x + 1\), and use this primitive polynomial to generate \(\mathbb{F}_2^6\). Let \(C\) be the code which consists of the spans of the following...
sixteen 3-subspaces:
\[
\langle \{\alpha^0, \alpha^1, \alpha^2\} \rangle, \quad \langle \{\alpha^2, \alpha^3, \alpha^{29}\} \rangle \\
\langle \{\alpha^0, \alpha^{15}, \alpha^{10}\} \rangle, \quad \langle \{\alpha^{25}, \alpha^0, \alpha^{58}\} \rangle \\
\langle \{\alpha^6, \alpha^{52}, \alpha^{51}\} \rangle, \quad \langle \{\alpha^1, \alpha^2, \alpha^{36}\} \rangle \\
\langle \{\alpha^{12}, \alpha^{54}, \alpha^{15}\} \rangle, \quad \langle \{\alpha^{36}, \alpha^{34}, \alpha^{30}\} \rangle \\
\langle \{\alpha^{10}, \alpha^{26}, \alpha^{25}\} \rangle, \quad \langle \{\alpha^6, \alpha^5, \alpha^{33}\} \rangle \\
\langle \{\alpha^{18}, \alpha^1, \alpha^{59}\} \rangle, \quad \langle \{\alpha^{19}, \alpha^{10}, \alpha^6\} \rangle \\
\langle \{\alpha^{33}, \alpha^{20}, \alpha^{59}\} \rangle, \quad \langle \{\alpha^{58}, \alpha^6, \alpha^{49}\} \rangle \\
\langle \{\alpha^{49}, \alpha^{26}, \alpha^{46}\} \rangle, \quad \langle \{\alpha^{36}, \alpha^{29}, \alpha^{26}\} \rangle
\]

3.3.3 Large non-sunflower equidistant codes

In this subsection we will consider the construction of the largest \( t \)-intersecting codes in \( G_q(n, k) \). For \( n \) large enough this code is a sunflower and hence for such large \( n \) we will consider also the largest \( t \)-intersecting code which is not a sunflower.

By Theorem 1 sunflowers are the largest constant dimension equidistant codes when the ambient space is large enough. The size of the largest sunflower is usually not known, but we know that it is equal to the size of a related partial spread. Therefore, we would like to know what is the size of the largest \( t \)-intersecting code in \( G_q(n, k) \) which is not a sunflower.

Assume we want to generate a \((k - r)\)-intersecting code in \( G_q(n, k) \). Clearly we must have \( n \geq k + r \). If \( k - r = 0 \) then any \((k - r)\)-intersecting code is a partial spread and hence also a sunflower. Therefore, we assume that \( k - r > 0 \). We start with the largest partial spread \( S \) in \( G_q(k + r, r) \). By Theorem 5 its size \( m \) is at least \( q^k + 1 \). \( S^\perp \) is a non-sunflower \((k - r)\)-intersecting code in \( G_q(k + r, k) \) whose size \( m \) is at least \( q^k + 1 \). If \( k - r > 1 \) then we don’t know how to construct a larger code. If \( k - r = 1 \) then larger codes of size \( \frac{q^{k+1}-1}{q-1} \) are constructed in subsection 3.3.1.

3.3.4 Equidistant codes in \( P_q(n) \)

So far we have considered only constant dimension equidistant codes. Can we get larger unrestricted subspace equidistant codes over \( \mathbb{F}_q^n \) than constant
dimension equidistant codes over $\mathbb{F}_q^n$? We start by considering first equidistant codes in the Hamming scheme. Let $B_q(n, d)$ be the maximum size of an equidistant code of length $n$ and minimum Hamming distance $d$ over $\mathbb{F}_q$. Let $B_q(n, d, w)$ be the maximum size of an equidistant code of length $n$, constant weight $w$, and minimum Hamming distance $d$. The following result, due to [22], shows that when discussing equidistant codes in the Hamming scheme, we may restrict our attention to constant weight codes:

**Theorem 16** $B_q(n, d) = 1 + B_q(n, d, d)$.

A related $q$-analog theorem might hold in some cases, but generally it does not hold as demonstrated in the following example. The example is specific in some sense, but it can be generalized to many other parameters.

Let $n$ be an odd integer for which the largest 2-intersecting equidistant code in $\mathcal{G}_2(n, 4)$ with minimum subspace distance 4 is a sunflower. The size of the largest partial 2-spread in $\mathcal{G}_2(n-2, 2)$ is $\frac{2^{n-2}-4}{3}$. Let $\mathcal{C}$ be such a partial spread. Clearly, $\mathcal{E}^2(\mathcal{C})$ is the largest 2-intersecting equidistant code in $\mathcal{G}_2(n, 4)$. Let $x, y, z,$ and $u$ be the only nonzero vectors of $\mathbb{F}_q^{n-2}$ which do not appear in any 2-subspace of $\mathcal{C}$. Let $v_1 = (0, 01), v_2 = (0, 10)$ be two vectors in $\mathbb{F}_q^2$ and let $\mathcal{E}^2(\mathcal{C}) \triangleq \{(X, 00) \cup \{v_1, v_2\} : X \in \mathcal{C}\}$, where $(X, 00) \triangleq \{(x, 00) : x \in X\}$. The code

$$\mathcal{C}' \triangleq \mathcal{E}^2(\mathcal{C}) \cup \{(0, 01), (y, 11), (0, 10), (z, 11), (0, 11), (x, 11)\}$$

is an equidistant code in $\mathcal{P}_2(n)$ whose size is $\frac{2^{n-2}+5}{3}$ and its subspace distance is 4. This code is larger than $\mathcal{E}^2(\mathcal{C})$, which implies that $q$-analog of Theorem [16] does not exist in general.

### 3.4 Equidistant Rank Metric Codes

In this section we present a connection between the construction presented in Section [3.3.1] and equidistant rank-metric codes. To the best of our knowledge, this is the first construction of an equidistant rank metric code whose matrices are not of full rank.

In this section we use a variant of the function $P$, defined in Section [3.3.1], denoted by $\varphi$. This variant may be considered as acting on matrices
from $\mathbb{F}_q^{k \times n}$ rather than on $\mathbb{G}_q(n, k)$, and maps them to $\mathbb{F}_q^{n \times k}$ rather than to $\mathbb{P}_q^{n \times k - 1}$.

**Definition 1** Given $M \in \mathbb{F}_q^{k \times n}$, identify the coordinates of $\mathbb{F}_q^{n \times k}$ with $k$-subsets of $[n]$, and define $\varphi(M)$ as a vector of length $n_k = \binom{n}{k}$ with:

$$(\varphi(M))_{i_1, \ldots, i_k} \triangleq \det M(i_1, \ldots, i_k)$$

where $M(i_1, \ldots, i_k)$ is the $k \times k$ sub-matrix of $M$ formed from columns $i_1, \ldots, i_k$ of $M$.

For $v, u \in \mathbb{F}_q^n$ denote $X_{v, u} \triangleq \varphi(u)$. For $v \in \mathbb{F}_q^n \setminus \{0\}$, define:

$$M_v \triangleq \begin{pmatrix} X_{v, e_1} \\ \vdots \\ X_{v, e_n} \end{pmatrix},$$

where $e_i$ is a unit vector of length $n$. Let $\mathbb{M} \triangleq \langle \{M_e\}_{i \in [n]} \rangle$. In the rest of this section we prove that $\mathbb{M} \setminus \{0\}$ is an equidistant constant rank code of size $q^n - 1$.

The following lemma shows that for $k = 2$, the function $\varphi$ is linear in some sense, when one of the vectors in the matrix it operates on is fixed.

**Lemma 4** If $u, v, w \in \mathbb{F}_q^n$ and $\alpha, \beta \in \mathbb{F}_q$ then $X_{v, \alpha u + \beta w} = \alpha \cdot X_{v, u} + \beta \cdot X_{v, w}$. Similarly, $X_{\alpha u + \beta w, v} = \alpha \cdot X_{u, v} + \beta \cdot X_{w, v}$.

**Proof.** Consider the $\{s, t\}$ coordinate of the vector $\alpha \cdot X_{v, u} + \beta \cdot X_{v, w}$:

$$(\alpha \cdot X_{v, u} + \beta \cdot X_{v, w})_{\{s, t\}} = \alpha (v_s u_t - v_t u_s) + \beta (v_s w_t - v_t w_s)$$

$$= v_s (\alpha u_t + \beta w_t) - v_t (\alpha u_s + \beta w_s)$$

$$= v_s (\alpha u + \beta w)_t - v_t (\alpha u + \beta w)_s$$

$$= \left( \varphi \left( \begin{pmatrix} \alpha u + \beta w \\ v \end{pmatrix} \right) \right)_{\{s, t\}}$$

$$= (X_{v, \alpha u + \beta w})_{\{s, t\}}.$$
Corollary 4 If \( V \in \mathcal{G}_q(n,1) \) and \( v \in V \setminus \{0\} \) then \( P_V = \left\langle \{X_v,z^i\}_{i=1}^{n-1} \right\rangle \), where \( z^i, i \in [n-1] \) was defined in Lemma 3.

We now show that each nonzero codeword in \( \mathcal{M} \) can be written as \( M_v \) for some \( v \in \mathbb{F}_q^n \), where \( \text{rank}(M_v) = n - 1 \). Hence, the linearity of the code implies that the rank of the difference between any two matrices in the code is \( n - 1 \), and therefore the code is equidistant.

Lemma 5 \( \mathcal{M} = \{ M_v | v \in \mathbb{F}_q^n \} \).

Proof. Let \( M = \sum_{i=1}^{n} \alpha_i M_{e_i} \in \mathcal{M} \), for some \( \alpha_i \in \mathbb{F}_q, i \in [n] \). By Lemma 4 the \( j \)th row of \( M \) is:

\[
\sum_{i=1}^{n} \alpha_i X_{e_i,e_j} = X_{(\sum_{i=1}^{n} \alpha_i e_i), e_j} = X_{v,e_j},
\]

where \( v = \sum_{i=1}^{n} \alpha_i e_i \), and therefore \( M = M_v \). Conversely, let \( v = \sum_{i=1}^{n} \alpha_i e_i \). The same arguments (in reversed order) shows that \( M_v = \sum_{i=1}^{n} \alpha_i M_{e_i} \) and hence \( M_v \in \mathcal{M} \).

Lemma 6 If \( v = \sum_{i=1}^{n} \alpha_i e_i, \alpha_i \in \mathbb{F}_q, i \in [n] \) is a non-zero vector in \( \mathbb{F}_q^n \) then \( \text{rank}(M_v) = n - 1 \).

Proof. By Corollary 4 the rows of \( M_v \) contain a basis for the codeword \( P_{(v)} \) (following the notations of Theorem 15). Recall (see Lemma 3) that a basis for \( P_{(v)} \) is obtained by omitting one vector from the set \( \{X_{v,e_i}\}_{i \in [n]} \). Note that \( r \) is the index of the omitted vector and \( \alpha_r \neq 0 \) (see Lemma 3). To complete the proof, we have to show that the vector \( X_{v,e_r} \) lies inside \( P_{(v)} \). Notice that by Lemma 3 we have that \( \alpha_r \neq 0 \). We have that

\[
\sum_{i \in [n] \setminus \{r\}} \alpha_i X_{v,e_i} = \varphi \left( \sum_{i \in [n] \setminus \{r\}} \alpha_i e_i \right)
= \varphi \left( \frac{v}{v - \alpha_r e_r} \right)
= \varphi \left( \frac{-\alpha_r}{-\alpha_r e_r} \right)
= -\alpha_r \varphi \left( \frac{v}{e_r} \right) = -\alpha_r X_{v,e_r}.
\]
Therefore, \( X_{v,e_r} = -\frac{1}{\alpha_r} \sum_{i \in [n] \setminus \{r\}} \alpha_i X_{v,e_i} \). Thus, \( \text{rowspan} (M_v) = P_{(v)} \) and \( \text{rank} (M_v) = \dim P_{(v)} = n - 1. \)

**Lemma 7** If \( M_v = M_u \) for \( u, v \in \mathbb{F}_q^n \) then \( u = v \).

**Proof.** If \( M_u = M_v \) then \( X_{v,e_j} = X_{u,e_j} \) for all \( j \in [n] \), which implies

\[
\forall j \in [n], \forall \{s, t\} \in \binom{[n]}{2}, \ t_s e_j,t - v_t e_j,s = u_s e_j,t - u_t e_j,s. \tag{3.3}
\]

In particular, for any \( i \in [n] \setminus \{j\} \) we may choose \( s = i, t = j \) and obtain that \( v_i = u_i \). Since (3.3) holds for any \( j \in [n] \), we also have \( v_j = u_j \) and thus \( u = v \). \( \blacksquare \)

**Corollary 5** \( |M| = q^n \).

**Corollary 6** \( M \setminus \{0\} \) is an equidistant constant rank code over \( \mathbb{F}_q \) with matrices of size \( n \times \binom{n}{2} \), rank \( n - 1 \), rank distance \( n - 1 \), and size \( q^n - 1 \).

There is some similarity between the code \( M \) and the Sylvester’s type Hadamard matrix of order \( 2^n \) [24]. This code is a binary linear of dimension \( n \) and length \( 2^n \). Each codeword, except for the all zeros codeword has weight \( 2^{n-1} \) and the mutual Hamming distance between any two codewords in \( 2^{n-1} \). The analog to the code \( M \) seems to be obvious.

### 3.5 Recursive Construction of Equidistant Subspace Codes

We have shown in Section 3.3.1 a direct construction of an 1-intersecting code in \( \mathcal{G}_q (n_2, n - 1) \). In this section we prove that this code can be constructed recursively, by using some of the results of Section 3.4.

Let \( \mathcal{C}_{n-1}, \mathcal{C}_n \) be the 1-intersecting codes in \( \mathcal{G}_q \left( \binom{n-1}{2}, n - 2 \right), \mathcal{G}_q \left( \binom{n}{2}, n - 1 \right) \), respectively, as constructed in Section 3.3.1. We first present a construction of some code \( \mathcal{D} \subseteq \mathcal{G}_q \left( \binom{n}{2}, n - 1 \right) \) from \( \mathcal{C}_{n-1} \) and later prove that \( \mathcal{D} = \mathcal{C}_n \).

Let \( \hat{v} \in \mathbb{F}_q^{n-1} \setminus \{0\} \). For the purpose of the construction, let \( X_{\hat{v},\hat{e}_i} = \varphi(\hat{v}) \) as in Section 3.4 (\( \hat{e}_i \) is a unit vector of length \( n - 1 \)), and let \( \mathcal{B}_{\hat{v}} = \{ \hat{z}^i \}_{i=1}^{n-2} \) be the set of \( n - 2 \) unit vectors of length \( n - 1 \) such that \( P_{(\hat{v})} = \left\langle \left\{ X_{\hat{v},\hat{z}^i} \right\}_{i \in \mathcal{B}_{\hat{v}}} \right\rangle \).
as denoted in the proof of Lemma 3. For \( v \in \mathbb{F}_q^n \setminus \{0\} \) we define \( e_i, X_{v, e_i} \) and \( B_v \) similarly.

For each codeword \( P(\hat{v}) \in \mathbb{C}_{n-1} \) we construct \( q \) codewords in \( \mathbb{D} \), denoted by \( \{U_{\hat{v}, a}\}_{a \in \mathbb{F}_q} \), as follows:

\[
\begin{align*}
U_{\hat{v},0} & \triangleq \text{rowspan} \begin{pmatrix} \hat{v} & 0 \\ 0 & X_{\hat{v}, \hat{z}^1} \\ \vdots & \vdots \\ 0 & X_{\hat{v}, \hat{z}^{n-2}} \end{pmatrix} \\
\forall a \neq 0, U_{\hat{v}, a} & \triangleq \text{rowspan} \begin{pmatrix} [ccc|c] \\ I_{(n-1) \times (n-1)} \\ \vdots \\ a \cdot X_{\hat{v}, \hat{e}_{n-1}} \end{pmatrix}
\end{align*}
\]

In addition, we add a codeword \( U_0 \triangleq \text{rowspan} (I_{(n-1) \times (n-1)} | 0) \) to \( \mathbb{D} \).

**Theorem 17** \( \mathbb{D} = \mathbb{C}_n \).

**Proof.** We prove that any \( P(v) \in \mathbb{C}_n \) is equal to some codeword in \( \mathbb{D} \). The equality, \( \mathbb{D} = \mathbb{C}_n \), will follow since \( |\mathbb{D}| \leq |\mathbb{C}_{n-1}| \cdot q + 1 = \left[ \frac{n-1}{1} \right] \cdot q + 1 = \left[ \frac{n}{1} \right] = |\mathbb{C}_n| \). Let \( P(v) \in \mathbb{C}_n \) for \( v \in \mathbb{F}_q^n \setminus \{0\} \), and let \( v = (a, \hat{u}) \neq (0, 0) \), where \( a \in \mathbb{F}_q, \hat{u} \in \mathbb{F}_q^{n-1} \). To find the related codeword in \( \mathbb{D} \), we distinguish between the following three cases:

**Case 1.** \( a = 0 \). W.l.o.g we choose \( B_v \) such that \( z^1 = e_1 \), where \( e_1 \) is a unit vector of length \( n \). We have that \( X_{v, z^1} = (-\hat{u}, 0) \), where \( 0 \) is the all zeros vector of length \( \left( \frac{n-1}{2} \right) \); for all \( z^i \in B_v, 2 \leq i \leq n-1 \) we have that \( X_{v, z^i} = (0, X_{\hat{u}, \hat{z}^i}) \), where \( \hat{z}^i \) is the \((n-1)\)-suffix of \( \hat{z}^i \) and \( 0 \) is the all zeros vector of length \( n-1 \). Hence, Corollary 4 implies that \( P(v) = \left\langle \{X_{v, z^i}\}_{i=1}^{n-1} \right\rangle = U_{\hat{u}, 0} \).

**Case 2.** \( a \neq 0, u \neq 0 \). For \( i \geq 2 \) we have:

\[
X_{v, e_i} = \left( \begin{array}{c}
0, \cdots, 0, a, 0, \cdots, 0 \\
\text{n-1 entries, (i-1)th equals a}
\end{array} \right), \quad X_{\hat{u}, \hat{e}_{i-1}} \left( \begin{array}{c}
\text{(n-1) entries}
\end{array} \right)
\]

where \( \hat{e}_{i-1} \) is the \((n-1)\)-suffix of the unit vector \( e_i \in \mathbb{F}_q^n \). Since \( a \neq 0 \) we may choose \( B_v = \{e_i\}_{i=2}^n \) and obtain by Corollary 4 that \( P(v) = \left\langle \{X_{v, e_i}\}_{i=2}^n \right\rangle = U_{a, a^{-1}} \).
Case 3. \( a \neq 0, u = 0 \). For \( i \geq 2 \) we have:

\[
X_{v,e_i} = \left( \begin{array}{cccc}
0, \ldots, 0, a, 0, \ldots, 0
\end{array} \right),
\]

\( n - 1 \) entries, \((i-1)\)th equals \( a \).

We may similarly choose \( B_v = \{ e_i \}_{i=2}^n \) and obtain \( P_{(v)} = U_0 \).

Starting from an 1-intersecting code in \( \mathcal{G}_q(3,2) \) which consists of all the two-dimensional subspaces of \( \mathbb{F}_q^3 \) we can obtain all the codes constructed in Section 3.3 recursively. We note that the initial condition consists exactly of all the lines of the projective plane of order \( q \).

3.6 Conclusion and Problems for Future Research

We have made a discussion on the size of the largest \( t \)-intersecting equidistant codes. The largest codes are known to be the trivial sunflowers. We discussed trivial codes and surveyed the known results in this direction. A construction of non-sunflower 1-intersecting codes in \( \mathcal{G}_q(n,k) \), \( n \geq \binom{k+1}{2} \), whose size is \( \left\lceil \binom{k+1}{1} q \right\rceil \), based on the Plücker embedding, is given. We showed that in at least one case there are larger non-sunflower 1-intersecting equidistant codes. Many important and usually very difficult problems remained for future research. We list herein a few.

1. Find the size of the largest partial spread for any given set of parameters.

2. Prove (or disprove) that the size of a non-sunflower \( t \)-intersecting constant dimension code of dimension \( k \), where \( t > 1 \) and \( k > t + 1 \), is at most \( \left\lceil \binom{k+1}{1} q \right\rceil \).

3. Identify the cases for which the size of a non-sunflower \( t \)-intersecting constant dimension code of dimension \( k \), where \( t > 1 \) and \( k > t + 1 \), is less than \( \left\lceil \binom{k+1}{1} q \right\rceil \).

4. Identify the cases for which the size of a non-sunflower 1-intersecting constant dimension code of dimension \( k \), where \( k > 2 \), is greater than \( \left\lceil \binom{k+1}{1} q \right\rceil \).
5. Find new constructions for non-sunflower $t$-intersecting constant dimension codes of dimension $k$, where $t \geq 1$, $k > t + 1$, whose size is larger than the codes obtained from partial spreads and their orthogonal codes.

6. Prove or disprove that the size of the largest equidistance code with subspaces distance $d$ in $P_q(n)$ depends on the size of the largest equidistance code with subspace distance $d$ in $G_q(n,k)$ for some $k$.

7. Find $1$-intersecting codes in $G_q(n,k)$ of size $\left[\frac{k+1}{1}\right]_q$, $k > 3$ and $n < \binom{k+1}{2}$.

8. Develop the theory of constant rank codes.

9. Find large equidistant rank metric codes.
Bibliography


[22] F. Fu, T. Kløve, Y. Luo, and V. K. Wei, On equidistant constant

[23] R. Koetter and F. R. Kschischang, Coding for errors and erasures
in random network coding, *IEEE Transactions on Information


ometry of Balls in the Grassmannian and List Decoding of Lifted
Gabidulin Codes, *Designs, Codes, and Cryptography*, 73:393–416,
2014.

[26] J. Rosenthal and A. L. Trautmann, Decoding of subspace codes,
a problem of schubert calculus over finite fields, *arXiv:1209.2887*,
2012.

[27] R. M. Roth, Maximum-rank array codes and their application
to crisscross error correction, *IEEE Transactions on Information

[28] D. Silva, F. R. Kschischang, and R. Koetter, A rank-metric ap-
proach to error control in random network coding, *IEEE Trans-

[29] S.-T. Xia and F.-W. Fu, Johnson type bounds on constant di-
mension codes, *Designs, Codes, and Cryptography*, 50:163–172,
2009.
Chapter 4

Subspace Polynomials and Cyclic Subspace Codes

Eli Ben-Sasson, Tuvi Etzion, Ariel Gabizon, and Netanel Raviv

Abstract

Subspace codes have received an increasing interest recently due to their application in error-correction for random network coding. In particular, cyclic subspace codes are possible candidates for large codes with efficient encoding and decoding algorithms. In this paper we consider such cyclic codes and provide constructions of optimal codes for which their codewords do not have full orbits. We further introduce a new way to represent subspace codes by a class of polynomials called subspace polynomials. We present some constructions of such codes which are cyclic and analyze their parameters.

4.1 Introduction

Let $\mathbb{F}_q$ be the finite field of size $q$, and let $\mathbb{F}_q^* \triangleq \mathbb{F}_q \setminus \{0\}$. For $n \in \mathbb{N}$ denote by $\mathbb{F}_{q^n}$ the field extension of degree $n$ of $\mathbb{F}_q$ which may be seen as the vector space of dimension $n$ over $\mathbb{F}_q$. By abuse of notation, we will not distinguish between these two concepts. Given a non-negative integer
in $k \leq n$, the set of all $k$-dimensional subspaces of $\mathbb{F}_{q^n}$ forms a Grassmannian space (Grassmannian in short) over $\mathbb{F}_q$, which is denoted by $G_q(n,k)$. The size of $G_q(n,k)$ is given by the well-known Gaussian coefficient $[n \choose k]_q$. The set of all subspaces of $\mathbb{F}_{q^n}$ is called the projective space of order $n$ over $\mathbb{F}_q$ [8] and is denoted by $\mathcal{P}_q(n)$. The set $\mathcal{P}_q(n)$ is endowed with the metric $d(U,V) = \dim U + \dim V - 2 \dim(U \cap V)$. A subspace code is a collection $C$ of subspaces from $\mathcal{P}_q(n)$. In this paper we will be mainly interested in constant dimension codes (called also Grassmannian codes), that is, $C \subseteq G_q(n,k)$ for some $k \leq n$.

Subspace codes and constant dimension codes have attracted a lot of research in the last eight years. The motivation was given in [12], where it was shown how subspace codes may be used in random network coding for correction of errors and erasures. This application of subspace codes renewed the interest in a wide variety of problems related to vector spaces [1, 6, 17, 20], particularly in constructions of large codes with error correction capability, efficient encoding algorithms for these codes, as well as efficient decoding algorithms.

In [12] a novel construction of large subspace codes using linearized polynomials (a.k.a. $p$-polynomials [18]) is presented. These codes were later shown [19] to be related to optimal rank-metric codes through an operation called lifting. These two techniques and some of their variants are the main known tools for constructing subspace codes.

It was previously suggested [4, 8, 13] that cyclic subspace codes may present a useful structure that can be applied efficiently for the purpose of coding. For a subspace $V \in G_q(n,k)$ and $\alpha \in \mathbb{F}_{q^n}^*$ we define the cyclic shift of $V$ as $\alpha V \triangleq \{\alpha v \mid v \in V\}$. The set $\alpha V$ is clearly a subspace of the same dimension as $V$. Two cyclic shifts are called distinct if they form two different subspaces. A subspace code $C$ is called cyclic if for every $\alpha \in \mathbb{F}_{q^n}^*$ and every $V \in C$ we have $\alpha V \in C$.

In [8, 13] several examples of optimal cyclic subspace codes with small dimension were found. In [4] an optimal code which also forms a $q$-analog of Steiner system was presented. This code has an automorphism group which is generated by a cyclic shift and the Frobenius mapping (known together also as a normalizer of a Singer subgroup [4, 11 pp. 187-188]). These codes raised the plausible conjecture that large cyclic codes may be constructed in any dimension. However, the current approaches for construction of sub-
space codes fall short with handling cyclic codes. In this paper we aim at establishing new general techniques for constructions of cyclic codes.

In [21] a thorough algebraic analysis of the structure of cyclic orbit codes is given. One class of such codes is the cyclic codes. However, no nontrivial construction is given. In [9] a construction of cyclic codes with degenerated orbit (of size less than $\frac{q^n-1}{q-1}$) is given. This construction produces a subcode of some codes in our work (see Section 4.3.2). Both [9] and [21] raised the following conjecture:

**Conjecture 2** For every positive integers $n, k$ such that $k < \frac{n}{2}$, there exists a cyclic code of size $\frac{q^n-1}{q-1}$ in $G_q(n,k)$ and minimum distance $2k-2$.

Notice that for $k > \frac{n}{2} + 1$, a minimum distance of $2k-2$ is clearly not possible. The original conjecture [21] considered $k \leq \frac{n}{2}$. However, an exhaustive search which was used in [9] proved that the conjecture is false for $n = 8, k = 4, q = 2$. When $k < \frac{n}{2}$, it appears that there is enough flexibility that many such codes exist, while for $k = \frac{n}{2}$, such a code might not exist. Its existence depends on the existence of a subspace which forms a structure similar to a difference set [4]. In this paper it is proved that this conjecture is true for a given $k$ and infinitely many values of $n$, along with several options for explicit constructions (see Theorem 20). In [9, 21] it was also pointed out that it is not known how to construct cyclic codes with multiple orbits. In the sequel we show that our techniques can be useful for this purpose (see Lemma 13 and Construction 1).

One of the tools in our constructions is the so-called subspace polynomials, which are a special case of linearized polynomials. Subspace polynomials form an efficient method of representing subspaces, from which one can directly deduce certain properties of the subspace which are not evident in some other representations. These objects were studied in the past for various purposes, e.g., bounds on list-decoding of Reed-Solomon and rankmetric codes [22], construction of affine dispersers [2], and finding an element of high multiplicative order in a finite field [5].

The rest of this paper is organized as follows. Section II will start with the known definition of subspace polynomials. We continue to analyze properties of the subspaces corresponding to the subspace polynomials, in particular we examine distance properties induced by cyclic and Frobenius shifts of
these subspaces. Based on these properties, in Section 4.3 we consider constructions of optimal cyclic codes with degenerate orbits, and cyclic codes with full orbits. The main goal in constructing cyclic codes is to obtain as many full orbits as possible in the code. This task will be left for future work. In this work we consider first the existence of cyclic codes with one full length orbit and cyclic codes with multiple full length orbits. Conclusions are given in Section 4.4.

4.2 Subspaces and their Subspace Polynomials

For the rest of this paper $k$ and $n$ will be positive integers such that $2 < k < n$, and we denote $|\ell| = q^\ell$. We begin by defining linearized polynomials and subspace polynomials.

**Definition 2** A linearized polynomial was defined by Ore [18] as follows:

\[ P(x) \triangleq a_k \cdot x^k + a_{k-1} \cdot x^{k-1} + \cdots + a_1 \cdot x^1 + a_0 \cdot x \]

where the coefficients are in the finite field $\mathbb{F}_{q^n}$.

Linearized polynomials have numerous applications in classic coding theory (e.g., [16, Chapter 4]). It is widely known that the roots of any linearized polynomial form a subspace in some extension of $\mathbb{F}_{q^n}$ (seen as a vector space over $\mathbb{F}_q$) and for every $V \in \mathcal{G}_q(k, n)$, the polynomial $\prod_{v \in V} (x - v)$ is a linearized polynomial [16, p. 118]. We will be particularly interested in linearized polynomials that have simple roots with respect to some field $\mathbb{F}_{q^n}$.

**Definition 3** [2, 3, 5, 22] A monic linearized polynomial $P$ with coefficients in $\mathbb{F}_{q^n}$ is called a subspace polynomial with respect to $\mathbb{F}_{q^n}$ if the following equivalent conditions hold:

1. $P$ divides $x^{|n|} - x$.
2. $P$ splits completely over $\mathbb{F}_{q^n}$ and all its roots have multiplicity 1.

From now on, we shall omit the notation of $\mathbb{F}_{q^n}$ whenever it is clear from context. The first two lemmas are trivial and well known. The simplicity of the roots of a subspace polynomial (and in particular, the simplicity of 0) gives rise to the following lemma.
Lemma 8 In any subspace polynomial, the coefficient of $x$ is non-zero. Conversely, every linearized polynomial with non-zero coefficient of $x$ is a subspace polynomial in its splitting field.

Proof. It is readily verified that 0 is a root of multiplicity 1 if and only if the coefficient of $x$ is non-zero. Therefore, if $P$ is a subspace polynomial, all of its roots are of multiplicity 1 (see Definition 3), including 0. On the other hand, if $Q$ is a linearized polynomial with a non-zero coefficient of $x$, then by [14, Theorem 3.50, p. 108], all the roots of $Q$ have multiplicity 1. ■

It also follows from Definition 3 that for a given $V \in \mathcal{G}_q(n, k)$ the polynomial $P_v(x) \triangleq \prod_{v \in V}(x - v)$ is the unique subspace polynomial whose set of roots is $V$, which leads to the following lemma.

Lemma 9 Two subspaces are equal if and only if their corresponding subspace polynomials are equal.

Example 1 Let $t$ be a positive integer such that $t|n$. It is known that $\mathbb{F}_{q^t}$ is a subfield (in particular, a subspace) of $\mathbb{F}_{q^n}$. The subspace polynomial of $\mathbb{F}_{q^t}$ is $P_{\mathbb{F}_{q^t}}(x) = x^t - x$.

The connection between linearized polynomials and subspace polynomial is given by the following two claims.

Theorem 18 [14, Theorem 3.50, p. 108] If $P$ is a linearized polynomial whose splitting field is $\mathbb{F}_{q^n}$, then each root of $P$ in $\mathbb{F}_{q^n}$ has the same multiplicity, which is a non-negative power of $q$, and the roots form a linear subspace of $\mathbb{F}_{q^n}$.

Lemma 10 If $P(x)$ is a linearized polynomial with a leading coefficient $^\dagger \alpha \neq 0$ and the splitting field of $P(x)$ is $\mathbb{F}_{q^n}$, then $P(x) = \alpha P_V(x)^{[t]}$ for some subspace $V$ in $\mathbb{F}_{q^n}$ and some $t \in \mathbb{N}$.

Proof. According to Theorem 18 all the roots of $P$ are of the same multiplicity $q^t$ for some $t \in \mathbb{N}$, and these roots form a subspace $V$ of $\mathbb{F}_{q^n}$.

\[ ^\dagger \text{The leading coefficient of a polynomial is the coefficient of the monomial with the highest degree.} \]
Hence,

\[ P(x) = \alpha \prod_{v \in V} (x - v)^{[\ell]} = \alpha \left( \prod_{v \in V} (x - v) \right)^{[\ell]} = \alpha P_V(x)^{[\ell]} . \]

\[ \blacksquare \]

In the sequel, we show several connections between the coefficients of subspace polynomials and properties of the respective subspaces. One of the main tools in our analysis is the difference between the indices of the two topmost non-zero coefficients:

**Definition 4** For \( V \in \mathcal{G}_q(n, k) \) and \( P_V(x) = x^{[k]} + \sum_{j=0}^t \alpha_j x^{[j]} \), where \( \alpha_i \neq 0 \), let \( \text{gap}(V) \triangleq k - i \).

As the following lemma illustrates, the gap of two subspaces induces a lower bound on their related distance.

**Lemma 11** If \( V \in \mathcal{G}_q(n, k_1) \) and \( U \in \mathcal{G}_q(n, k_2) \) are two distinct subspaces such that \( k_1 \leq k_2 \) and

\[ P_V(x) = x^{[k_1]} + \sum_{j=0}^t \alpha_j x^{[j]} \]

\[ P_U(x) = x^{[k_2]} + \sum_{j=0}^s \beta_j x^{[j]} , \]

such that \( \alpha_i \neq 0 \) and \( \beta_s \neq 0 \), then \( \dim(U \cap V) \leq \max(s, t + k_2 - k_1) \).

**Proof.** According to the properties of \( \mathbb{F}_{q^n} \), for all \( \alpha, \beta \in \mathbb{F}_{q^n} \) and for all \( i \in \mathbb{N} \) we have that \( (\alpha + \beta)^{[i]} = \alpha^{[i]} + \beta^{[i]} \), and therefore

\[ P_V(x)^{[k_2-k_1]} = x^{[k_2]} + \sum_{j=0}^t \alpha_j^{[k_2-k_1]} x^{[j+k_2-k_1]} . \]

Since the polynomials \( P_V, P_V^{[k_2-k_1]} \) have the same set of roots, and since the roots of \( P_U \) are simple, it follows that \( \text{gcd}(P_V, P_U) = \text{gcd}(P_V^{[k_2-k_1]}, P_U) . \)

\[ ^2 \text{gcd}(s, t) \text{ stands for the greatest common denominator of the elements } s, t. \]
Hence, if \( Q(x) \triangleq P_U(x) - P_V(x)^{k_2-k_1} \) then

\[
\gcd(P_V, P_U) = \gcd(P_V^{k_2-k_1}, P_U) = \gcd(P_V^{k_2-k_1}, P_U(\mod P_V^{k_2-k_1})) = \gcd(P_V^{k_2-k_1}, Q(\mod P_V^{k_2-k_1})).
\]

Since \( \deg Q \leq \max([s], [t + k_2 - k_1]) \), it follows that

\[
\log_q \deg \gcd(P_V^{k_2-k_1}, Q(\mod P_V^{k_2-k_1})) \leq \max(s, t + k_2 - k_1).
\]

Since clearly, \( \gcd(P_U, P_V) = P_U \cap V \), it follows that \( \dim(U \cap V) \leq \max(s, t + k_2 - k_1) \). □

A special case of Lemma 11, where the subspaces \( U \) and \( V \) are of the same dimension \( k \), provides the following useful corollaries.

**Corollary 7** If \( U, V \in G_q(n, k) \) then \( \dim(U \cap V) \leq k - \min(\text{gap}(U), \text{gap}(V)) \).

**Corollary 8** If \( U, V \in G_q(n, k) \) then \( d(U, V) \geq 2 \min(\text{gap}(U), \text{gap}(V)) \).

**Remark 2** Corollary 8 is not tight, i.e., there exists subspaces \( U, V \in G_q(n, k) \) where \( \text{gap}(V) = \text{gap}(U) = 1 \) and \( d(U, V) = 2k - 2 \). For example, let \( \gamma \) be a root of \( x^7 + x + 1 = 0 \), and use this primitive polynomial to generate \( \mathbb{F}_{27} \). The following polynomials are subspace polynomials of \( U, V \in G_2(7, 3) \) for which \( \text{gap}(U) = \text{gap}(V) = 1 \) and \( d(U, V) = 2 \cdot 3 - 2 \cdot 1 = 4 \). In particular, \( U \) and \( V \) are cyclic shifts of each other.

\[
P_U(x) = x^3 + x^2 + (\gamma^6 + \gamma^4 + \gamma^3 + \gamma + 1)x^1 + (\gamma^3 + \gamma^2 + \gamma + 1)x
\]

\[
P_V(x) = x^3 + (\gamma^2 + 1)x^2 + (\gamma + \gamma + 1)x^1 + (\gamma^5 + \gamma^4 + \gamma)x
\]

Aside from cyclic shifts we will also use the well known Frobenius mapping \( F^i \) as a method to increase the size of the codes. For an element \( \alpha \in \mathbb{F}_{q^n} \) and \( i \in \{0, \ldots, n-1\} \), the \( \mathbb{F}_q \)-mapping \( F^i \) is defined as \( F^i(\alpha) = \alpha^{[i]} \) (see [13] p. 75)). For a subspace \( V \) and \( i \in \{0, \ldots, n-1\} \) the \( i \)th Frobenius shift of \( V \) is defined as \( F^i(V) \triangleq \{v^{[i]} \mid v \in V\} \). Since the function \( F^i \) is an
automorphism, it follows that the set $F^i(V)$ is a subspace of the same dimension as $V$. We now characterize the subspace polynomials of the subspaces resulting from these mappings.

**Lemma 12** If $V \in \mathcal{G}_q(n,k)$ and $\alpha \in \mathbb{F}_q^*$ then $P_{\alpha V}(x) = \alpha [k] \cdot P_V(\alpha^{-1} x)$. That is, if $P_V(x) = x^{[k]} + \sum_{j=0}^i \alpha_j x^{[j]}$ then $P_{\alpha V}(x) = x^{[k]} + \sum_{j=0}^i \alpha_j \alpha_j x^{[j]}$.

**Proof.** By definition,

$$P_{\alpha V}(x) = \prod_{u \in \alpha V} (x - u) = \prod_{v \in V} (x - \alpha v) = \alpha^{[k]} \prod_{v \in V} (\alpha^{-1} x - v) = \alpha^{[k]} \cdot P_V(\alpha^{-1} x) = x^{[k]} + \sum_{j=0}^i \alpha^{[k]-[j]} \alpha_j x^{[j]}.$$

**Lemma 13** If $V \in \mathcal{G}_q(n,k)$ and $P_V(x) = x^{[k]} + \sum_{j=0}^i \alpha_j x^{[j]}$ then for all $s \in \{0, \ldots, n-1\}$, $P_{F^s V}(x) = x^{[k]} + \sum_{j=0}^i F^s(\alpha_j) x^{[j]}$.

**Proof.** If $s \in \{0, \ldots, n-1\}$ and $u \in F^s(V)$ then $u = F^s(v)$ for some $v \in V$. Since $F^s$ is an automorphism, it follows that

$$u^{[k]} + \sum_{j=0}^i F^s(\alpha_j) u^{[j]} = F^s(v^{[k]} + \sum_{j=0}^i F^s(\alpha_j) F^s(v)^{[j]} = F^s(v^{[k]} + \sum_{j=0}^i F^s(\alpha_j v^{[j]} = F^s \left( v^{[k]} + \sum_{j=0}^i \alpha_j v^{[j]} \right) = F^s (P_V(v)) = F^s \left( \prod_{w \in V} (v - w) \right) = F^s(0) = 0.$$
Therefore all elements of \( F^s(V) \) are roots of \( x^k + \sum_{j=0}^i F^s(\alpha_j)x^j \). Since the degree of this polynomial is \( k \), the claim follows.

The next lemma shows a connection between the coefficients of the subspace polynomial of a given subspace \( V \in \mathcal{G}_q(n, k) \) and the number of its distinct cyclic shifts. To formulate our claim, we need the following equivalence relation.

**Definition 5** For \( \alpha, \beta \in \mathbb{F}_{q^n}^* \) and an integer \( t \) which divides \( n \), the equivalence relation \( \sim_t \) is defined as follows

\[
\alpha \sim_t \beta \iff \frac{\alpha}{\beta} \in \mathbb{F}_{q^t}.
\]

Clearly, if \( \alpha \sim_t \beta \) then \( \alpha \in \beta \mathbb{F}_{q^t}^* \cap \alpha \mathbb{F}_{q^t}^* \), and since all the cyclic shifts of \( \mathbb{F}_{q^t}^* \) in \( \mathbb{F}_{q^n}^* \) are disjoint, it follows that \( \beta \mathbb{F}_{q^t}^* = \alpha \mathbb{F}_{q^t}^* \). Hence, the equivalence classes under this relation are all the cyclic shifts of \( \mathbb{F}_{q^t}^* \) in \( \mathbb{F}_{q^n}^* \). Therefore, there are exactly \( \frac{q^n-1}{q^t-1} \) equivalence classes of \( \sim_t \), each of which is of size \( q^t - 1 \).

**Lemma 14** Let \( V \in \mathcal{G}_q(n, k) \) and \( P_V(x) = x^k + \sum_{j=0}^i \alpha_j x^j \). If \( \alpha_s \neq 0 \) for some \( s \in \{1, \ldots, i\} \) and \( \gcd(s, n) = t \) then \( \alpha V \neq \beta V \) for all \( \alpha, \beta \in \mathbb{F}_{q^n}^* \) such that \( \alpha \sim_t \beta \).

**Corollary 9** Let \( V \in \mathcal{G}_q(n, k) \) and \( P_V(x) = x^k + \sum_{j=0}^i \alpha_j x^j \). If \( \alpha_s \neq 0 \) for some \( s \in \{1, \ldots, i\} \) with \( \gcd(s, n) = t \) then \( V \) has at least \( \frac{q^n-1}{q^t-1} \) distinct cyclic shifts.

To construct codes with more than one orbit using the Frobenius automorphism, one would like to find a sufficient condition that a certain Frobenius shift is not a cyclic shift. Such a condition can be derived for the special case, where the subspace polynomial is a certain trinomial.

**Lemma 15** If \( V \in \mathcal{G}_q(n, k) \) and \( P_V(x) = x^k + \alpha_1 x^{[1]} + \alpha_0 x \), where \( \alpha_1 \neq 0 \), then there exists \( \alpha \in \mathbb{F}_{q^n}^* \), \( i \in \{0, \ldots, n-1\} \) such that \( F^i(V) = \alpha V \) if and only if

\[
\left( \frac{\alpha_0^{q^{i+1}}}{\alpha_1^{q^i-1}} \right)^{q^i-1} = 1.
\]

The proofs of Lemma 14 and Lemma 15 are given in .
4.3 Cyclic Subspace Codes

In this section some constructions of cyclic subspace codes are provided. We distinguish between two cases. In Subsection 4.3.1 we discuss codes whose codewords have a full length orbit. In Subsection 4.3.2 codes whose codewords do not have a full length orbit are discussed.

Definition 6 Given a subspace \( V \in \mathcal{G}_q(n,k) \), the set \( \{ \alpha V | \alpha \in \mathbb{F}_{q^n}^* \} \) is called the orbit of \( V \). The subspace \( V \) has a full length orbit if \( |\{ \alpha V | \alpha \in \mathbb{F}_{q^n}^* \}| = \frac{q^n - 1}{q - 1} \). If \( V \) does not have a full length orbit then it has a degenerate orbit.

Note, that a cyclic code with a full length orbit cannot have a minimum distance \( 2k \). This is a simple observation from the fact that each element \( \alpha \in \mathbb{F}_{q^n}^* \) appears in exactly \( \frac{q^k - 1}{q - 1} \) codewords.

We will give several simple related results on subspaces and the size of their orbits. The first claim may be extracted from [9, Corollary 3.13]. For completeness we include a shorter self-contained proof.

Lemma 16 If \( V \in \mathcal{G}_q(n,k) \) then \( |\{ \alpha V | \alpha \in \mathbb{F}_{q^n}^* \}| = \frac{q^n - 1}{q - 1} \) for some \( t \) which divides \( n \).

Proof. Let \( \gamma \) be a primitive element in \( \mathbb{F}_{q^n} \) and let \( \ell \in \mathbb{N} \) be the smallest integer such that \( \gamma^{\ell}V = V \). Clearly, \( \ell | q^n - 1 \) and it is readily extracted that each \( i \in \mathbb{N} \) and each \( 0 \leq s < \ell \) satisfy \( \gamma^s V = \gamma^{i \ell + s} V \). Furthermore, for every \( s_1, s_2 \in \{0, \ldots, \ell - 1\} \) the sets \( A_{s_1} = \{ \gamma^{i \ell + s} | i \in \mathbb{N} \} \) satisfy \( |A_{s_1}| = |A_{s_2}| \). Let \( \gamma^{i_1 \ell}, \gamma^{i_2 \ell} \in A_0 \) for some \( i_1, i_2 \in \mathbb{N} \). Since \( A_0 = \{ \gamma^{i \ell} | i \in \mathbb{N} \} \) \( (= \{ \gamma^m | \gamma^m V = V \}) \) it follows that

\[
\left( \gamma^{i_1 \ell} + \gamma^{i_2 \ell} \right) V \subseteq \gamma^{i_1 \ell} V + \gamma^{i_2 \ell} V = V + V = V,
\]

and hence \( \gamma^{i_1 \ell} + \gamma^{i_2 \ell} \in A_0 \), that is, \( A_0 \) is closed under addition. Since \( A_0 \) is also closed under multiplication, it follows that \( A_0 \) is the multiplicative group of some subfield \( \mathbb{F}_{q^t} \) of \( \mathbb{F}_{q^n} \). Therefore, \( |\{ \alpha V | \alpha \in \mathbb{F}_{q^n}^* \}| = \ell = \frac{q^n - 1}{q - 1} \).

An immediate consequence of Lemma 16 is that the largest possible size of an orbit is \( \frac{q^n - 1}{q - 1} \), which justifies Definition 6. As will be shown in the sequel.
(see Section 4.3.2), the parameter $t$ from Lemma 16 must also divide $k$. A formula for the number of orbits of each possible size is given in [7]. Most of the $k$-dimensional subspaces of $\mathbb{F}_{q^n}$ have full length orbits. The main goal in constructing cyclic codes is to obtain as many orbits as possible in the code. This task will be left for future work. In this work we consider first the existence of cyclic codes with one full length orbit and cyclic codes with multiple full length orbits. Later, we consider the largest cyclic codes for which all the orbits are degenerate.

4.3.1 Codes with Full Length Orbits

**Lemma 17** [14, p. 91, Theorem 3.20], [16, p. 107, Theorem 11] The polynomial $Q(x) \triangleq x^n - x$ is the product of all monic irreducible polynomials over $\mathbb{F}_q$ with degree dividing $n$.

**Theorem 19** If $q^k - 1$ divides $n$ and $x^{[k]} - 1 + x^{[1]} - 1 + 1$ is irreducible over $\mathbb{F}_q$ then the polynomial $x^{[k]} + x^{[1]} + x$ is a subspace polynomial with respect to $\mathbb{F}_q^n$.

**Proof.** Assume that $x^{[k]} - 1 + x^{[1]} - 1 + 1$ is irreducible over $\mathbb{F}_q$ and its degree divides $n$. By Lemma 17, $x^{[k]} - 1 + x^{[1]} - 1 + 1 | Q(x)$, and hence $x^{[k]} + x^{[1]} + x | Q(x)$. Therefore, $x^{q^k} + x^q + x$ is a subspace polynomial (see Definition 3), i.e., $P_V(x) = x^{[k]} + x^{[1]} + x$ for some subspace $V$. ■

**Corollary 10** If $q^k - 1$ divides $n$, $x^{[k]} - 1 + x^{[1]} - 1 + 1$ is irreducible over $\mathbb{F}_q$, and $V \in G_q(n, k)$ is the subspace whose subspace polynomial is $x^{[k]} + x^{[1]} + x$, then $C \triangleq \{ \alpha V | \alpha \in \mathbb{F}_{q^n}^* \}$ is a cyclic subspace code of size $q^n - 1$ and minimum distance $2k - 2$.

**Proof.** According to Corollary 9 since the coefficient of $x^q$ in $P_V$ is nonzero, there are $q^n - 1$ distinct cyclic shifts in $C$. By Lemma 12 and Corollary 8 the minimum distance of $C$ is at least $2k - 2$. As observed before, a cyclic code with a full length orbit cannot have a minimum distance $2k$ since each $\alpha \in \mathbb{F}_{q^n}^*$ appears in exactly $q^{k-1}$ codewords. Hence, the minimum distance is exactly $2k - 2$. ■

Although there exists an extensive research on irreducible trinomials over finite fields (e.g., [23]), no explicit construction of irreducible trinomials of the above form is known. However, the following examples were easily found using a computer search.
Example 2 Since the polynomials $x^{2^k-1} + x + 1$ are irreducible over $\mathbb{F}_2$ for all $k \in \{2, 3, 4, 6, 7, 15\}$, it follows that the polynomial $x^{2^k} + x^2 + x$ is a subspace polynomial of a subspace $V \in \mathcal{G}_2 ((2^k - 1)t, k)$ for all $t \in \mathbb{N}$. Therefore, the code $C \triangleq \{ \alpha V \mid \alpha \in \mathbb{F}_{q^{(2^k-1)t}}^* \}$ is a cyclic code of size $2^t(2^k-1) - 1$ and minimum distance $2k - 2$ in $\mathcal{G}_2 ((2^k - 1)t, k)$.

By using a similar approach we have that for any $k$ and $q$, cyclic codes in $\mathcal{G}_q (n, k)$ can be explicitly constructed for infinitely many values of $n$. The construction will make use of the following lemma.

Lemma 18 If $f(x) = \prod_{i=1}^{t} p_i^{s_i}(x)$ is a polynomial over $\mathbb{F}_q$ and $p_1(x), \ldots, p_t(x)$ are its irreducible factors in $\mathbb{F}_q$ then $f(x)$ splits completely in $\mathbb{F}_q^n$ for $n = \text{lcm}\{\deg p_i(x)\}_{i=1}^{t}$.

Proof. According to [14, Corollary 2.15, p. 52], the splitting field of an irreducible polynomial of degree $m$ over $\mathbb{F}_q$ is $\mathbb{F}_{q^m}$. Therefore, for each $i = 1, \ldots, t$, the splitting field of $p_i(x)$ is $\mathbb{F}_{q^{n_i}}$, where $n_i \triangleq \deg p_i$. For any $i$, the only finite fields that contain $\mathbb{F}_{q^{n_i}}$ are of the form $\mathbb{F}_{q^r}$ for $r$ such that $n_i|r$. Hence, the smallest field that contains $\mathbb{F}_{q^{n_i}}$ for all $i$ is $\mathbb{F}_{q^n}$. ■

Theorem 20 For any $k$ and $q$ we may explicitly construct a cyclic subspace code of size $\frac{q^n - 1}{q-1}$ and minimum distance $2k - 2$ in $\mathcal{G}_q (n, k)$ for infinitely many values of $n$.

Proof. By factoring $T(x) \triangleq x^{[k]} + x^{[1]} + x$ and computing the least common multiple of the degrees of its factors we find the degree of the splitting field of $T(x)$ (see Lemma 18). The subspace $V$, whose corresponding subspace polynomial is $T(x)$, may be easily found by finding the kernel of the linear transformation defined by $T$. If $C \triangleq \{ \alpha V \mid \alpha \in \mathbb{F}_q^* \}$ then by Corollary 8 there are $q^n - 1$ distinct cyclic shifts in $C$. By Lemma 12 and Corollary 8 the minimum distance of $C$ is least $2k - 2$, and hence it is exactly $2k - 2$. Infinitely many values of $n$ are given by the degrees of all extensions fields of the splitting field of $T(x)$. ■

Remark 3 Note that Corollary 10 is a special case of Theorem 20, which proves Conjecture 2 for infinitely many values of $n$.

\[^3\text{lcm}\{s_i\}_{i=1}^t\] stands for the least common multiplier of the integers $s_1, \ldots, s_t$. 52
Remark 4 The codes implied by Theorem 19 and Theorem 20 cannot be enlarged using the Frobenius isomorphism due to Lemma 15, since for any \( i \in \{0, \ldots, n - 1\} \) we have that the \( i \)th Frobenius shift is also a cyclic shift.

Now, we present a general method for constructing cyclic codes in \( \mathcal{G}_q(N, k) \), where \( N = t \cdot n \) for some prime \( n \), which have more than one full length orbit. We do so by using the Frobenius automorphism.

Let \( N = t \cdot n \) and let \( \gamma \) be a primitive element in \( \mathbb{F}_{q^N} \). Note, that the set \( \{0\} \cup \{\gamma^{(q^N-1)/(q^n-1)}\}^{n-2} \) is the unique subfield \( \mathbb{F}_{q^n} \) of \( \mathbb{F}_{q^N} \). Let \( V \) be a subspace of \( \mathbb{F}_{q^n} \). Since \( \mathbb{F}_{q^n} \subseteq \mathbb{F}_{q^N} \) we can view the subspace \( V \) as a subspace of \( \mathbb{F}_{q^N} \) over \( \mathbb{F}_q \).

Lemma 19 Let \( n \) be a prime, \( n \mid N \), \( V \in \mathcal{G}_q(N, k) \) and \( P_V(x) = x^k + \alpha_0 x + \alpha_1 x^1 \), where \( \alpha_0, \alpha_1 \in \mathbb{F}_{q^n}^* \). If \( \alpha_1^{q^k-1} \sim_1 \alpha_0^{q^k-1} \) (see Definition 5) then the code \( C \subseteq \mathcal{G}_q(N, k) \) defined by

\[
C \triangleq \bigcup_{i=0}^{n-1} \left\{ \alpha \cdot F^i(V) \mid \alpha \in \mathbb{F}_{q^N}^* \right\}
\]

(4.1)

is of size \( n \cdot \frac{q^N-1}{q-1} \) and minimum distance \( 2k - 2 \).

The following lemma shows that the coefficients \( \alpha_0, \alpha_1 \) from Lemma 19 may be easily found in \( \mathbb{F}_{q^n} \).

Lemma 20 Let \( n \) be prime and let \( \gamma \) be a primitive element in \( \mathbb{F}_{q^n} \). If \( \alpha_0 \triangleq \gamma \) and \( \alpha_1 \triangleq \gamma^q \) then \( \alpha_1^{q^{k-1}} \approx_1 \alpha_0^{q^{k-1}} \).

As a consequence of Lemma 19 and Lemma 20, whose proofs are deferred to , we have the following theorem.

Theorem 21 Let \( n \) be prime, \( \gamma \) a primitive element of \( \mathbb{F}_{q^n} \), and define \( \alpha_0 \triangleq \gamma \) and \( \alpha_1 \triangleq \gamma^q \). If \( \mathbb{F}_{q^N} \) is the splitting field of the polynomial \( x^k + \alpha_1 x^1 + \alpha_0 x \) and \( V \in \mathcal{G}_q(N, k) \) its corresponding subspace, then

\[
C \triangleq \bigcup_{i=0}^{n-1} \left\{ \alpha \cdot F^i(V) \mid \alpha \in \mathbb{F}_{q^N}^* \right\}
\]

is a cyclic code of size \( n \cdot \frac{q^N-1}{q-1} \) and minimum distance \( 2k - 2 \).
Note that the construction in Theorem 21 improves the construction of Theorem 20. In Theorem 20 we construct a code with one full length orbit, where in Theorem 21 we add multiple orbits without compromising the minimum distance.

4.3.2 Codes with degenerate orbits

In this subsection it is shown that certain sets of subspaces of $G_q(n,k)$ that may be considered as subspaces over a subfield (larger than $F_q$) of $F_q^n$, form a cyclic code with a unique subspace polynomial structure. The cyclic property and the minimum distance of this code are an immediate consequence of this unique structure. In this subsection we will distinguish between $F_q^n$, $F_q^n$, and $F_{q^d}$, where $d$ is an integer which divides $n$.

Let $g$ be any $F_{q^d}$ isomorphism between $F_{q^d}$ and $F_q^n$. Notice that for all $u,v \in F_{q^d}$ and $\alpha,\beta \in F_{q^d}$, we have $g(\alpha v + \beta u) = \alpha g(v) + \beta g(u)$. For $V \in G_{q^d}(n/d,k/d)$ let $G(V) \triangleq \{g(v) | v \in V\}$. The set $G(V)$ is clearly a subspace of dimension $k$ over $F_q$ in $F_q^n$. Furthermore, the function $G : G_{q^d}(n/d,k/d) \rightarrow G_q(n,k)$ is injective since $g$ is injective.

**Construction 1** For $n,k \in \mathbb{N}$ and $d \in \mathbb{N}$ such that $d | \gcd(n,k)$, let $C_d$ be the code

$$\{G(V) | V \in G_{q^d}(n/d,k/d)\}.$$

Since $C_d$ is the image of an injective function from $G_{q^d}(n/d,k/d)$ to $G_q(n,k)$, we have that

**Corollary 11** $|C_d| = \left[\frac{n/d}{k/d}\right]_{q^d}$.

**Remark 5** The code $C_d$ from construction 1 may be alternatively defined as

$$C_d \triangleq \left\{ \sum_{i=1}^{k/d} \alpha_i F_{q^d} \mid \alpha_1,\ldots,\alpha_{k/d} \in F_{q^n} \text{ are linearly independent over } F_{q^d} \right\}.$$  

The code $C_d$ may also be defined as the set of all subspaces of $G_q(n,k)$ that are also subspaces over $F_{q^d}$. If $k|n$ and $d = k$ then Construction 1 is the well-known construction of spread codes as the cyclic shifts of the subfield $F_{q^d}$. The proofs of these facts, which are not used in the sequel, are left as an exercise to the reader.
Note that subspace polynomials of subspaces in $G_q(n, k)$ and in $G_{q^d}(n/d, k/d)$ are both defined over $F_{q^n} = F_{q^{dn/d}}$. The subspaces in $C_d$ admit a unique subspace polynomial structure, from which the useful properties of $C_d$ are apparent.

**Lemma 21** If $V \in G_q(n, k)$ then $V \in C_d$ if and only if $P_V(x) = \sum_{i=0}^{k/d} c_i x^{[di]}$ for some $c_i$’s in $F_{q^n}$.

**Proof.** Let $V \in C_d$, and let $U \in G_{q^d}(n/d, k/d)$ be such that $G(U) = V$ (see Construction 1). By Definition 3 it follows that $P_U(x)(q^d)^{n/d} - x$. Since $x(q^d)^{n/d} - x = x^n - x$, it follows that $P_U$ is a subspace polynomial of a subspace $W \in G_q(n, k)$. The roots of $P_U$ are precisely the set $\{g(u) | u \in U\}$, where $g$ is the isomorphism between $F_{q^{n/d}}$ and $F_{q^n}$, and hence, $W = V$. Since $P_U$ is a subspace polynomial of a subspace in $G_{q^d}(n/d, k/d)$, it is of the form $P_U(x) = \sum_{i=0}^{k/d} c_i x^{(q^d)^i}$. Since $P_V = P_U$, the claim follows.

Conversely, let $V \in G_q(n, k)$ with $P_V(x) = \sum_{i=0}^{k/d} c_i x^{[di]}$. By Definition 3, it follows that $P_V|_{x^{[n]} - x}$, and thus $P_V|_{x(q^d)^{n/d} - x}$. Therefore $P_V$ is a subspace polynomial of some $U \in G_{q^d}(n/d, k/d)$, and hence $V \in C_d$. ■

**Corollary 12** $C_d \subseteq G_q(n, k)$ is a cyclic subspace code.

**Proof.** Let $V \in C_d$ and $\alpha \in F_{q^n}^*$. By Lemma 21 the subspace polynomial of $V$ is of the form $P_V(x) = \sum_{i=0}^{k/d} c_i x^{[di]}$ for some $c_i \in F_{q^n}$. By Lemma 12 the subspace polynomial of $\alpha V$ is $P_V(x) = \sum_{i=0}^{k/d} c_i \alpha x^{[di]}$. Again by Lemma 21 it follows that $\alpha V \in C_d$. ■

Since for $V \in C_d$ we have that gap$(V) \geq d$, the following result is a consequence of Corollary 8 and Definition 4.

**Corollary 13** The minimum distance of $C_d$ is $2d$.

**Remark 6** Some of the above properties of $C_d$ can be proved without using subspace polynomials. However, since one of the main goals of this paper is to present the concept of subspace polynomials, these properties are proved using subspace polynomials.

The structure of the subspace polynomials of the codewords of $C_d$ allows us to construct a code $C$ which is a union of $C_{d_i}$ for distinct $d_i$’s which divide gcd$(n, k)$. We now analyze the size and distance of the resulting code.
Lemma 22 Let $k, n \in \mathbb{N}, k < n$. If $d_1, \ldots, d_t$ divide both $n$ and $k$ and $d = \text{lcm}(d_1, \ldots, d_t)$ then $\bigcap_{i=1}^t C_{d_i} = C_d$.

Proof. According to Lemma 21 if $V \in C_d$ then $P_V(x) = \sum_{i=0}^{k/d} c_i x^{id}$. Since $d_j | d$ for each $j$, we may also write $P_V(x) = \sum_{i=0}^{k/d_j} c'_i x^{id_j}$, where all additional coefficients are 0, and thus $V \in C_{d_j}$ for each $j$.

On the other hand, if $V \in \bigcap_{i=1}^t C_{d_i}$, again by Lemma 21 it follows that all nonzero coefficients of $P_V$ correspond to $x^{\ell}$ such that $d_j | \ell$ for each $j$. Thus, $d | \ell$ and $V \in C_d$. ■

Construction 2 Let $k, n \in \mathbb{N}, k < n$. If $d_1, \ldots, d_t$ divide both $n$ and $k$ then let $C \triangleq \bigcup_{i=1}^t C_{d_i}$.

Lemma 23 $C$ is a cyclic code with codewords of dimension $k$ and minimum distance $2 \min_{i=1}^t \{d_i\}$. The size of $C$ is given by

$$|C| = \sum_{i=1}^t \left[ \frac{n}{d_i} \right]_{q^{d_i}} - \sum_{i<j} \left[ \frac{n}{\text{lcm}(d_i, d_j)} \right]_{q^{\text{lcm}(d_i, d_j)}} + \sum_{i<j<\ell} \left[ \frac{n}{\text{lcm}(d_i, d_j, d_\ell)} \right]_{q^{\text{lcm}(d_i, d_j, d_\ell)}} - \ldots .$$

Proof. By Corollary 13 we have that gap($V$) $\geq \min_{i=1}^t \{d_i\}$ for each $V \in C$, and hence the minimum distance of $C$ is at least $2 \min_{i=1}^t \{d_i\}$ by Corollary 8. By Corollary 11 we have that $|C_{d_i}| = \left[ \frac{n}{d_i} \right]_{q^{d_i}}$ for each $i$. Furthermore, by Lemma 22 the size of the intersection of $C_{d_1}, \ldots, C_{d_\ell}$ is $\left[ \frac{n}{d} \right]_{q^d}$ where $d = \text{lcm}(d_1, \ldots, d_\ell)$. These facts allow us to obtain the exact size of $C$ using the inclusion-exclusion principle [13 Chapter 10]. ■

Using similar techniques, we can show that a cyclic code over a large field may be embedded in a Grassmannian over a smaller field, while preserving cyclicity and multiplying the minimal distance by some factor. Note that Construction 1 is a special case of this technique, where the embedded code is $G_q^{n/d} (n/d, k/d)$.

Theorem 22 Let $d$ be an integer such that $d | \gcd(n, k)$. If $C \subseteq G_q^{n/d} (n/d, k/d)$
is a cyclic code with minimum distance \(2 \cdot (k/d) - 2\delta\) then there exists a cyclic code \(C' \subseteq G_q(n, k)\) of size \(|C|\) and minimum distance \(2k - 2d\delta\).

**Proof.** Let \(g : \mathbb{F}_{q/d}^n \rightarrow \mathbb{F}_q^n\) and \(G : \mathbb{G}_{q/d}(n/d, k/d) \rightarrow \mathbb{G}_q(n, k)\) be the embeddings defined earlier in this subsection. If \(C' \triangleq \{G(V) | V \in C\}\) then \(|C'| = |C|\), since \(G\) is injective. The cyclic property of \(C'\) follows from the fact that \(P_V(x) = P_{G(V)}(x)\) for all \(V \in C\), as shown in the proof of Lemma 21. To bound the minimum distance of \(C'\) it suffices to show that if \(U_1, U_2 \in C\) then

\[
\dim (G(U_1) \cap G(U_2)) = d \cdot \dim(U_1 \cap U_2).
\]

Indeed, if \(w \triangleq \dim(U_1 \cap U_2)\), then since \(g\) is an isomorphism of subspaces over \(\mathbb{F}_{q/d}\), it follows that the set \(Z \triangleq \{g(z) | z \in U_1 \cap U_2\}\) is a subspace of \(\mathbb{F}_q^n\) over \(\mathbb{F}_q\). By a simple counting argument, \(\dim Z = dw\), and hence, \(\dim (F(U_1) \cap F(U_2)) \geq dw\). Assuming for contradiction that \(\dim (F(U_1) \cap F(U_2)) > dw\) clearly implies that \(\dim(U_1 \cap U_2) > w\), a contradiction. 

### 4.4 Conclusions and Future Work

In this paper we have considered constructions of cyclic subspace codes. We have proved the existence of a cyclic code in \(\mathbb{G}_q(n, k)\) for any given \(k\) and infinitely many values of \(n\). The constructed codes have minimum subspace distance \(2k - 2\), the normalizer of a Singer subgroup is their automorphism group if \(n\) is a prime, and they have full length orbits for all values of \(n\). We have also constructed large codes when all the orbits are degenerated. We have shown how the representation of subspaces by their subspace polynomials can be used in constructing subspace codes.

For future research, the main problems are to construct cyclic codes of large size, to explore the structure and properties of our codes, and to examine possible decoding algorithms for them. It is easily verified that the vast majority of subspaces have full length orbits. Therefore, it seems reasonable to conjecture that full length orbits with minimum distance \(2k - 2\) exist for any value of \(n, k, q\) (see Conjecture 2). Although the codes presented in Section 4.3.1 are the first known explicit construction of such codes, they are most likely the tip of the iceberg, and codes of these parameters are abound.
Although the gap of two polynomials provides significant information about the intersection of their respective subspaces, Remark 2 shows that the gap might not be the most efficient tool for this purpose. Therefore, another open problem is finding a better measure for the intersection of two subspaces, and in particular, two subspaces from the same orbit.

A prominent part of the study of subspace polynomials relies on understanding the connection between the coefficients of a polynomial and the size of the respective splitting field. Hence, any progress in this direction may provide an improvement of our results.

**Acknowledgments**

The authors would like to thank the anonymous reviewers for their useful comments. Especially, the comments of one of the reviewers helped to amend the paper. The authors also want to thank Thomas Honold for bringing [7] to their attention.
Bibliography


1165-1173, 2011.

codes and stabilizer subfields,” *Advances in Mathematics of Com-
munications*, vol. 9, no. 2, pp. 177–197, 2015.

ematics: A Foundation for Computer Science,” *Addison-Wesley*,
1994

1967.

in random network coding, *IEEE Transactions on Information

sion codes with a prescribed minimum distance,” *Lecture Notes

Mathematics and Its Applications*, Cambridge University Press,


[16] F.J. MacWilliams, and N.J. Sloane, “The Theory of Error-

[17] W. J. Martin, and X. J. Zhu, “Anticodes for the Grassman and
bilinear forms graphs,” *Designs, Codes and Cryptography*, vol. 6,

Appendix A

Proof. (of Lemma 14) Assume for contradiction that \( \alpha V = \beta V \) for some \( \alpha, \beta \in \mathbb{F}_q^* \), where \( \alpha \nsim \beta \). By Lemma 12

\[
P_{\alpha V}(x) = x^{[k]} + \sum_{j=0}^{i} \alpha_j \cdot \alpha^{[k]-[j]} x^{[j]}
\]

\[
P_{\beta V}(x) = x^{[k]} + \sum_{j=0}^{i} \alpha_j \cdot \beta^{[k]-[j]} x^{[j]}.
\]

The equality \( \alpha V = \beta V \), together with Lemma 9 imply that

\[
\begin{cases}
\alpha_s \alpha^{[k]-[s]} = \alpha_s \beta^{[k]-[s]} \\
\alpha_0 \alpha^{[k]-1} = \alpha_0 \beta^{[k]-1}
\end{cases}
\]
and since $\alpha_0 \neq 0$ by Lemma 8 it follows that

$$
\begin{align*}
\begin{cases}
(\frac{\alpha}{\beta})^{[k]-[s]} = 1 \\
(\frac{\alpha}{\beta})^{[k]-1} = 1
\end{cases}
\end{align*}
$$

By dividing the second equation by the first equation, we get $(\frac{\alpha}{\beta})^{[s]-1} = 1$. Hence, $\text{ord}(\frac{\alpha}{\beta})|\gcd(q^n - 1, q^s - 1)$. It is well known that in $\mathbb{Z}_{q^n-1}$, $\gcd(q^n - 1, q^s - 1) = q^{\gcd(n,s)} - 1$ (e.g., [10, p. 147, s. 38]). Therefore, $\text{ord}(\frac{\alpha}{\beta})|q^{\gcd(n,s)} - 1$, which implies that $\frac{\alpha}{\beta} \in \mathbb{F}_q$ since $t = \gcd(n, s)$, and hence $\alpha \sim_t \beta$, a contradiction.

\section*{Proof. (of Lemma 15)}
Assume $F^i(V) = \alpha V$ for some $\alpha$. By Lemmas 12 and 13

$$
\begin{align*}
P_{\alpha V}(x) &= x^{[k]} + \alpha^{[k]-[1]} \cdot \alpha_1 x^{[1]} + \alpha^{[k]-1} \cdot \alpha_0 x \\
P_{F^i(V)}(x) &= x^{[k]} + F^i(\alpha_1) x^{[1]} + F^i(\alpha_0) x.
\end{align*}
$$

By Lemma 9

$$
\begin{align*}
\begin{cases}
\alpha^{[k]-[1]} \cdot \alpha_1 = F^i(\alpha_1) \\
\alpha^{[k]-1} \cdot \alpha_0 = F^i(\alpha_0)
\end{cases}
\end{align*}
$$

$$
\begin{align*}
\begin{cases}
\alpha^{[k]-[1]} \cdot \alpha_1 = \alpha_1^{[i]} \\
\alpha^{[k]-1} \cdot \alpha_0 = \alpha_0^{[i]}
\end{cases}
\end{align*}
$$

Since $\alpha_0 \neq 0$ (by Lemma 8) and $\alpha_1 \neq 0$, $\alpha^{[1]-1} = \left(\frac{\alpha_0}{\alpha_1}\right)^{q^t-1}$. Using some
algebraic manipulations we have,

\[
\alpha^{[k]−[1]} = \alpha_1^{[i]−1}
\]

\[
(q-1) \left( \frac{q^k-q}{q-1} \right) = \alpha_1^{q^{i−1}}
\]

\[
\left( \frac{\alpha_0}{\alpha_1} \right) \left( q^i-1 \right) \left( \frac{q^k-q}{q-1} \right) = \alpha_1^{q^{i−1}}
\]

\[
\frac{q^k-q}{q-1} \cdot (q^{i−1}) = 1
\]

\[
\frac{\alpha_0}{\alpha_1} \left( \frac{q^k-q}{q-1} \right) (q^i-1) = 1
\]

\[
\frac{\alpha_0}{\alpha_1} \left( \frac{q^{k-1}}{q-1} \right) (q^i-1) = 1
\]

\[
\left( \frac{\alpha_0}{\alpha_1} \right) \left( \frac{q^k-q}{q-1} \right) q^{i−1} = 1,
\]

which concludes the proof of one direction of the lemma. Now assume

\[
\left( \frac{\alpha_0}{\alpha_1} \right) q^{i−1} = 1.
\]
Define $\alpha \triangleq \left( \frac{\alpha_0}{\alpha_1} \right)^{q^i-1}$. We get
\[
\begin{pmatrix}
\frac{q^k-q}{\alpha_0^{q^i-1}} \\
\frac{q^k-1}{\alpha_1^{q^i-1}}
\end{pmatrix}^{q^i-1} = 1
\]
\[
\begin{pmatrix}
\frac{q^k-q}{\alpha_0^{q^i-1}} \\
\frac{q^k-1}{\alpha_1^{q^i-1} \cdot \alpha_1}
\end{pmatrix}^{q^i-1} = 1
\]
\[
\begin{pmatrix}
\frac{q^{i-1}}{\alpha_0^{q^i-1}} \\
\frac{q^{i-1}}{\alpha_1^{q^i-1}}
\end{pmatrix}^{q^k-q} = \alpha_1^{q^i-1}
\]
\[
a^{q^k-q} = \alpha_1^{q^i-1}.
\]
In addition, we have $\alpha^{q^k-1} = \alpha^{q^k-q} \cdot \alpha^{q^i-1} = \left( \alpha_1^{q^i-1} \right) \cdot \left( \frac{\alpha_0^{q^i-1}}{\alpha_1^{q^i-1}} \right) = \alpha_0^{q^i-1}$.
Therefore:
\[
\begin{cases}
\alpha^{q^k-q} = \alpha_1^{q^i-1} \\
\alpha^{q^k-1} = \alpha_0^{q^i-1} \\
\alpha^{q^k-q} \cdot \alpha_1 = \alpha_1^{q^i} \\
\alpha^{q^k-1} \cdot \alpha_0 = \alpha_0^{q^i},
\end{cases}
\]
which implies that $F^i(V) = \alpha V$ due to equality between the coefficients of the corresponding subspace polynomials.

\section*{Appendix B}

\textbf{Proof.} (of Lemma 19) The code $C$ is obviously cyclic. By Lemmas 12, 13 and Corollary 8, the dimension of the intersection between any two distinct subspaces in $C$ is at most 1, and hence the minimum distance of $C$ is $2k - 2$.

To show that $|C| = n \cdot \frac{q^N-1}{q-1}$, fix $i$ and notice that by Lemma 13 we have that the coefficient of $x^{[1]}$ in $P_{F^i(V)}(x)$ is non-zero. Therefore, Corollary 9 implies that the set $\{ \alpha \cdot F^i(V) \mid \alpha \in \mathbb{F}_q^* \}$ consists of $\frac{q^N-1}{q-1}$ distinct subspaces.

To complete the proof, we have to show that all the sets in the union
in \((4.1)\) are disjoint. Let \(i, j \in \{0, \ldots, n - 1\}, i \neq j,\) and assume for contradiction that there exists \(\beta, \gamma \in \mathbb{F}_{q^N}^*\) such that \(\beta F^i(V) = \gamma F^j(V).\) W.l.o.g assume that \(j > i,\) and denote \(U \triangleq F^i(V).\) Notice that by Lemma 13 we have

\[
P_U(x) = P_{F^i(V)}(x) = x^{[k]} + F^i(\alpha_1) \cdot x^{[l]} + F^i(\alpha_0) \cdot x
\]

\[
= x^{[k]} + \alpha_1^{[q]} \cdot x^{[l]} + \alpha_0^{[q]} \cdot x.
\]

Since \(F^{j-i}(U) = \frac{\beta}{\gamma} \cdot U,\) we may apply Lemma 15 to get

\[
\left(\frac{\alpha_0^{q^{j-i}}} {\alpha_1^{q^{j-i-1}}}\right)^{q^i} = 1.
\]

(4.2)

Denote \(z \triangleq \frac{\alpha_0^{q^{j-i-1}}}{\alpha_1^{q^{j-i-1}}}\) and notice that

A1. Equation \((4.2)\) implies \(z^{q^i(q^{j-i-1})} = 1.\)

A2. The condition \(\alpha_1^{q^{j-i-1}} \approx_1 \alpha_0^{q^{j-i-1}}\) implies \(z \notin \mathbb{F}_q.\)

A3. Since \(\alpha_0, \alpha_1 \in \mathbb{F}_{q^n}^*\) it follows that \(z \in \mathbb{F}_{q^n}^*\).

By A1 and A3 we have that \(\text{ord}(z)\) divides both \(q^j(q^{j-i} - 1)\) and \(q^n - 1,\) therefore \(\text{ord}(z) | \gcd(q^j(q^{j-i} - 1), q^n - 1).\) Since \(q^n - 1\) is not a power of \(q,\) it follows that \(\gcd(q^n - 1, q^j) = 1,\) and hence,

\[
\gcd(q^j(q^{j-i} - 1), q^n - 1) = \gcd(q^{j-i} - 1, q^n - 1).
\]

It is well known that in any field \(\gcd(x^r - 1, x^s - 1) = x^{\gcd(r, s)} - 1\) (e.g., \([10,\ p. 147,\ s. 38]\)). Therefore, the primality of \(n\) implies that \(\gcd(q^{j-i} - 1, q^n - 1) = q^{\gcd(j-i, n)} - 1 = q - 1,\) and hence \(\text{ord}(z) | q - 1.\) The only elements of \(\mathbb{F}_{q^N}\) whose order divides \(q - 1\) are the elements of \(\mathbb{F}_q,\) and hence \(z \in \mathbb{F}_q,\) a contradiction to A2. ■
Proof. (of Lemma 20) Assume for contradiction that
\[ \frac{q^{k-q}}{q^{n-1}} \sim_{1} \frac{q^{k-1}}{q^{n-1}}, \]
i.e., there exists \( \alpha \in \mathbb{F}_{q}^{*} \) such that
\[ \alpha \cdot \gamma^{q^{k-q}} = (\gamma^{q})^{q^{k-1}}. \tag{4.3} \]
Raising both sides of (4.3) by the \((q-1)\)th power yields
\[ \gamma^{q^{k}-q} = \gamma^{q^{k+1}-q} \]
\[ \gamma^{q^{k}(q-1)} = 1. \tag{4.4} \]
Since \( q \in \mathbb{Z}_{q^{n}-1}^{*} \), it follows that \( q \) has a multiplicative inverse \( w \) modulo \( q^{n}-1 \). By raising both sides of (4.4) by the \( w^{k} \)th power we get that \( \gamma^{q-1} = 1 \), and hence, \( \gamma \in \mathbb{F}_{q} \), a contradiction. \( \blacksquare \)
Chapter 5

Some Gabidulin Codes Cannot Be List Decoded Efficiently at any Radius

Netanel Raviv and Antonia Wachter-Zeh

Disclaimer

After this paper was published, a mistake was found in the proof of Theorem 25, simultaneously and separately by Mr. Netanel Raviv and Prof. Pierre Loidreau. A detailed explanation regarding the mistake and its correction is given in Subchapter 5.A. Albeit slightly weaker after the correction, the main statement of this paper still holds in its entirety, that is, there exist Gabidulin codes which cannot be list decoded efficiently at any radius.

Abstract

Gabidulin codes can be seen as the rank-metric equivalent of Reed–Solomon codes. It was recently proven, using subspace polynomials, that Gabidulin codes cannot be list decoded beyond the so-called Johnson radius. In another result, cyclic subspace codes were constructed by inspecting the connection between subspaces and their subspace polynomials. In this paper, these subspace codes are used to prove two bounds on the list size in decoding
certain Gabidulin codes. The first bound is an existential one, showing that exponentially-sized lists exist for codes with specific parameters. The second bound presents exponentially-sized lists explicitly, for a different set of parameters. Both bounds rule out the possibility of efficiently list decoding several families of Gabidulin codes for any radius beyond half the minimum distance. Such a result was known so far only for non-linear rank-metric codes, and not for Gabidulin codes. Using a standard operation called lifting, identical results also follow for an important class of constant dimension subspace codes.

5.1 Introduction

Rank-metric codes have recently attracted increasing interest due to their application to error correction in random network coding [30] where they can be used to construct constant dimension subspace codes. Further applications of codes in the rank metric include cryptography [11, 19], space-time coding [20, 21] and distributed storage systems [28, 29].

For a prime power $q$, let $\mathbb{F}_q$ be the field with $q$ elements. For an integer $n$, let $\mathbb{F}_q^n$ be the extension field of degree $n$ of $\mathbb{F}_q$ (which may be seen as the vector space of dimension $n$ over $\mathbb{F}_q$, denoted by $\mathbb{F}_q^n$), and $\mathbb{F}_q^* \triangleq \mathbb{F}_q^\ast \setminus \{0\}$. For $m \geq n$, a rank-metric code is a set of $m \times n$ matrices over $\mathbb{F}_q$, or alternatively, a set of vectors of length $n$ over the extension field $\mathbb{F}_q^m$, where the distance between two matrices is the rank of their difference. The rate of a rank metric code of size $M$ is $\log_q M \cdot \frac{mn}{\log_q M}$. Gabidulin codes, introduced by [6, 10, 27], may be seen as the rank-metric equivalent of Reed–Solomon codes. These codes are defined as evaluations of linearized polynomials (see below) of bounded degree at a given set of linearly independent evaluation points. We note that Gabidulin codes, and rank-metric codes in general, may be defined for any $m \geq n$, while our results only apply for the case $n$ divides $m$ (and in some cases, when $n + 1$ divides $m$ by puncturing). In particular, our results apply for $n = m$.

Given a word $w \in \mathbb{F}_q^n$ (or alternatively, a matrix $w \in \mathbb{F}_q^{m \times n}$), a list decoding algorithm outputs all Gabidulin codewords that are inside a ball of radius $\tau$, centered at $w$, where $\tau$ is possibly larger than the unique decoding radius of the code. For a given code, a natural question to ask is: for which values of $\tau$ can list decoding be done efficiently? List decoding of rank-
metric codes and Gabidulin codes was recently studied in [7, 15, 31]. In [31], it was shown that Gabidulin codes cannot be list decoded beyond the Johnson radius. This result was generalized to any rank-metric code by [7]. When $m$ is sufficiently large, [7] also showed that with high probability a random rank-metric code can be efficiently list decoded. Further, it was shown in [31] that there is no Johnson-like polynomial upper bound on the list size since there exists a non-linear rank-metric code with exponentially growing list size for any radius greater than the unique decoding radius. In [15], an explicit subcode of a Gabidulin code was shown to be efficiently list decodable. In addition, [7, 15, and 31] have noted that it is not known if Gabidulin codes themselves can be efficiently list decoded beyond the unique decoding radius. In this paper, it is shown that the answer to this question is negative.

Clearly, if there exists a word $w \in \mathbb{F}_{q^m}^n$ with exponentially many Gabidulin codewords in a radius $\tau$ around it, then efficient list decoding is not possible for this radius. This combinatorial technique was used in [4] to show the limits of list decoding of Reed–Solomon codes, and in [31] to show the limits of list decoding of Gabidulin codes.

The main tool in [4, 31] is subspace polynomials, which are a special type of linearized polynomials. Linearized polynomials, defined by Ore [24], are polynomials of the form

$$P(x) = a_r \cdot x^r + \cdots + a_1 \cdot x^1 + a_0 \cdot x,$$

where $[i] \triangleq q^i$ and the coefficients are in the finite field $\mathbb{F}_{q^n}$ for some given $n$. For a linearized polynomial $P$, define the $q$-degree of $P$ as $\deg_q P \triangleq r = \log_q \deg P$. Using the isomorphism between $\mathbb{F}_{q^n}$ and $\mathbb{F}_q^n$, every linearized polynomial may be seen as an $\mathbb{F}_q$-linear function from $\mathbb{F}_q^n$ to itself [18, Chapter 4, p. 108], that is, for every $\alpha, \beta \in \mathbb{F}_q$ and $u, v \in \mathbb{F}_q^n$, each linearized polynomial $P$ satisfies $P(\alpha v + \beta u) = \alpha P(v) + \beta P(u)$. A subspace polynomial is defined as follows.

**Definition 7** [2, 3, 4, 5, 31] A monic linearized polynomial $P$ is called a subspace polynomial with respect to $\mathbb{F}_{q^n}$ if it satisfies the following equivalent conditions:

**A1.** $P$ divides $x^n - x$. 

69
A2. $P$ splits completely over $\mathbb{F}_{q^n}$ and all its roots have multiplicity one.

A3. For some $0 \leq r \leq n$, there exists an $r$-dimensional subspace $V$ of $\mathbb{F}_{q^n}$ such that $P(x) = \prod_{v \in V} (x - v)$.

By A3, each subspace $V$ corresponds to a unique subspace polynomial, denoted $P_V$. Subspace polynomials are an efficient method of representing subspaces, from which one can directly deduce certain properties of the subspace which are not evident in some other representations. These objects were studied in the past for various other purposes, e.g., construction of affine dispersers [3], finding an element of high multiplicative order in a finite field [5], and construction of cyclic subspace codes [2]. Albeit this wide range of applications, not much is known about the coefficients of subspace polynomials and their connection to the properties of the subspace.

It is known that all roots of every linearized polynomial have the same multiplicity, which is an integer power of $q$, and these roots form a subspace in the extension field [18, Theorem 3.50, p. 108]. Therefore, any monic linearized polynomial is a power of a subspace polynomial with respect to its splitting field. However, the structure of the coefficients of subspace polynomials, compared to other linearized polynomials of the same degree, is generally not known. A partial answer to this question was given by [2], and we use similar techniques to show limits of list decoding of Gabidulin codes.

Ben-Sasson et al. [4] proved that a given set of subspace polynomials with mutual top coefficients provides an upper bound on the list decoding radius of Reed–Solomon codes. A counting argument was later applied in order to show that such large sets of subspace polynomials do exist. A similar technique was used in [31] to show the limits of list decoding of Gabidulin codes. In the sequel, the existence of a set of subspaces whose polynomials have a larger agreement is proved (Theorem 25). This set is a subset of a subspace code by [2]. Furthermore, explicit dense sets of words in a Gabidulin code are provided (Theorem 26). Both bounds are used to show that the respective families of Gabidulin codes cannot be list decoded efficiently at all. That is, there exist received words that have exponentially many codewords around them, already for a radius which is only larger than the unique decoding radius by one (Examples 3 and 4 and Theorem 26). Due to a technical limitation of our techniques, the presented families have
rate at least $\frac{1}{5}$.

Subspace codes have attracted an increasing interest recently due to their application in error correction in random network coding [17]. It is widely known that rank-metric codes are deeply connected to constant dimension subspace codes through an operation called lifting [12, 30]. This operation preserves the distance and the cardinality of the original rank-metric code. An important family of nearly optimal constant dimension subspace codes are *lifted Gabidulin codes* (that are a special case of the so-called Kötter and Kschischang codes [17]), which result from Gabidulin codes by lifting (see Definition 10). List decoding of subspace codes was extensively studied in recent years. In particular, several variants and subcodes of the Kötter and Kschischang codes were shown to be efficiently list decodable (e.g., [7, 15, 16, 22, 23] and references therein), and bounds equivalent to [31] were discussed in [26]. Our results about Gabidulin codes also apply for lifted Gabidulin codes, and thus we get families of subspace codes that cannot be list decoded efficiently at any radius. Our techniques may also be used for showing limits to list decoding of Reed–Solomon codes, but the resulting bounds are too weak to provide any useful insight.

These results reveal a significant difference in list decoding Gabidulin and Reed–Solomon codes, although the definitions of these code classes strongly resemble each other. Namely, Reed–Solomon codes can be efficiently list decoded up to the Johnson radius (with the Guruswami–Sudan algorithm [14]), whereas we have just proven that (some classes of) Gabidulin codes cannot be list decoded efficiently at all.

The rest of the paper is organized as follows. Notations for subspace codes and the subspace code from [2] will be described in Section 5.2 together with the required background on cyclic shifts of subspaces and $q$-associates of polynomials. In Section 5.3 the code from Section 5.2 is used to prove the existence of a certain set of subspace polynomials, and the notion of $q$-associates is used to show an explicit set of another type of subspace polynomials. The improved bounds on list decodability of Gabidulin codes are discussed in Section 5.4, implications about subspace codes are discussed in Section 5.5 and conclusions are given in Section 5.6. A discussion about the inapplicability of our techniques to list decodability of Reed–Solomon codes appears in.
5.2 Preliminaries

The set $\mathcal{G}_q(n,r)$, called the Grassmannian, is the set of all subspaces of dimension $r$ ($r$-subspaces, in short) of $\mathbb{F}_q^n$. The size of $\mathcal{G}_q(n,r)$ is given by the Gaussian coefficient $\begin{bmatrix} n \\ r \end{bmatrix}_q \triangleq \prod_{i=0}^{r-1} \frac{q^n - q^i}{q^r - q^i}$, which satisfies $q^r(n-r) \leq \begin{bmatrix} n \\ r \end{bmatrix}_q \leq 4q^{r(n-r)}$.[12] A constant dimension subspace code [17] is a subset of $\mathcal{G}_q(n,r)$ under the subspace metric $d_S(U,V) = \dim U + \dim V - 2\dim(U \cap V)$.

An extensively used concept in this paper is cyclic shifts of subspaces, defined as follows.

**Definition 8** For $V \in \mathcal{G}_q(n,r)$ and $\alpha \in \mathbb{F}_q^*$ let

$$\alpha V \triangleq \{\alpha v | v \in V\}.$$ 

The set $\alpha V$, which is clearly a subspace of the same dimension as $V$, is called a cyclic shift of $V$. Cyclic shifts were shown to be useful for constructing subspace codes [2][9]. The set of all cyclic shifts of $V \in \mathcal{G}_q(n,r)$ is called the orbit of $V$, and its size is $\frac{q^n-1}{q^r-1}$ for some integer $t$ which divides $n$. The size of the orbit and the structure of its subspace polynomials can be derived by inspecting the subspace polynomial of $V$, as shown in the following lemmas.

**Lemma 24** [2] Lemma 5] If $V \in \mathcal{G}_q(n,r)$ and $\alpha \in \mathbb{F}_q^*$ then $P_{\alpha V}(x) = \alpha^{[r]} \cdot P_V(\alpha^{-1}x)$. That is, if $P_V(x) = x^{[r]} + \sum_{j=0}^{r-1} \alpha_j x^{[j]}$ then $P_{\alpha V}(x) = x^{[r]} + \sum_{j=0}^{r-1} \alpha^{[r]-[j]} \alpha_j x^{[j]}$.

**Lemma 25** [2] Corollary 3] Let $V \in \mathcal{G}_q(n,r)$ and $P_V(x) = x^{[r]} + \sum_{j=0}^{r-1} \alpha_j x^{[j]}$. If $\alpha_s \neq 0$ for some $s \in \{1, \ldots, r-1\}$ and $\gcd(s,n) = t$, then $V$ has at least $\frac{q^n-1}{q^r-1}$ distinct cyclic shifts.

In [2] it is shown that subspaces in $\mathcal{G}_q(n,r)$, that may be considered as subspaces over a subfield of $\mathbb{F}_q^n$ which is larger than $\mathbb{F}_q$, admit a unique subspace polynomial structure. In what follows we cite the essentials from [2]. For an integer $g$ such that $g | \gcd(n,r)$, let $h$ be any $\mathbb{F}_{q^g}$ isomorphism between $\mathbb{F}_{q^g}^n$ and $\mathbb{F}_{q^g}$, and notice that for all $u, v \in \mathbb{F}_{q^g}^n$ and $\alpha, \beta \in \mathbb{F}_{q^g}$, we have that $h(\alpha v + \beta u) = \alpha h(v) + \beta h(u)$. For $V \in \mathcal{G}_{q^g}(n/g, r/g)$ let $H(V) \triangleq \{h(v) | v \in V\}$. The set $H(V)$ is clearly a subspace of dimension $r$ over $\mathbb{F}_q$ in $\mathbb{F}_{q^g}$. Furthermore, the function $H : \mathcal{G}_{q^g}(n/g, r/g) \rightarrow \mathcal{G}_q(n,r)$ is injective since $h$ is injective.
Construction 3 [2] Construction 1] For integers \( g, n, \) and \( r \) such that \( 0 < r < n \) and \( g|\gcd(n,r) \), let

\[
\mathbb{C}_g \triangleq \{ H(V) | V \in \mathcal{G}_{q^g}(n/g,r/g) \}.
\]

Clearly, for \( g = 1 \) Construction 3 is trivial. Thus, we henceforth assume that \( g \geq 2 \), i.e., \( n \) and \( r \) have a non-trivial \( \gcd \). The subspace code \( \mathbb{C}_g \) has minimum subspace distance \( 2g \), and it may alternatively be defined as direct sums of cyclic shifts of \( \mathbb{F}_{q^g} \) or as the set of all subspaces of \( \mathcal{G}_q(n,r) \) that are subspaces over \( \mathbb{F}_{q^g} \) as well [2]. Since \( \mathbb{C}_g \) is the image of an injective function from \( \mathcal{G}_{q^g}(n/g,r/g) \) to \( \mathcal{G}_q(n,r) \), we have the following.

Corollary 14 [2, Corollary 5] \(|\mathbb{C}_g| = \left[ \frac{n}{g} \right] \left[ \frac{r}{g} \right] q^g \).

The subspaces in \( \mathbb{C}_g \) admit a unique subspace polynomial structure, from which the results in this paper follow.

Lemma 26 [2] Lemma 14] If \( V \in \mathcal{G}_q(n,r) \) then \( V \in \mathbb{C}_g \) if and only if \( P_V(x) = \sum_{i=0}^{r/g} c_i x^{[g]} \), where \( c_i \in \mathbb{F}_{q^g}, \forall i \in \{0, \ldots, r/g\} \).

Another concept used in our constructions is the notion of \( q \)-associates. Two polynomials over \( \mathbb{F}_{q^g} \) of the form \( \ell(x) = \sum_{i=0}^{d} \alpha_i x^i \) and \( L(x) = \sum_{i=0}^{d} \alpha_i x^{q^i} \), are called \( q \)-associates of each other. For any \( g \in \mathbb{N} \), one can similarly define \( q^g \)-associativity, where \( \ell(x) = \sum_{i=0}^{d} \alpha_i x^i \), and \( L(x) = \sum_{i=0}^{d} \alpha_i x^{q^{gi}} \) are \( q^g \)-associates of each other. Linearized polynomials over \( \mathbb{F}_q \) are deeply connected to their \( q \)-associates as follows.

Lemma 27 [18, Theorem 3.62, p. 116] If \( L_1(x) \) and \( L(x) \) are linearized polynomials over \( \mathbb{F}_q \) with \( q \)-associates \( \ell_1(x) \) and \( \ell(x) \), then \( L_1(x) \) divides \( L(x) \) if and only if \( \ell_1(x) \) divides \( \ell(x) \).

5.3 Sets of Subspaces Polynomials with Mutual Top Coefficients

In [4] (resp. [31]) it was shown that sets of subspace polynomials that agree on many of their top coefficients provide a bound on the list decodability of Reed–Solomon (resp. Gabidulin) codes. By Lemma 26 it is evident that all subspace polynomials of subspaces in \( \mathbb{C}_g \) agree on their topmost \( g \) coefficients.
Using a counting argument we may prove the existence of a subset of $C_g$ whose corresponding subspace polynomials agree on a larger number of top coefficients.

**Theorem 23** If $g, n,$ and $r$ are integers such that $0 < r < n$, $g \mid \gcd(r, n)$, and $\ell$ is the unique non-negative integer such that $r = n - g(\ell + 1)$, then there exists a subset of $C_g$ of size at least

$$\frac{\binom{n/g}{r/g} q^n}{q^{n\ell}},$$

whose subspace polynomials agree on their topmost $g(\ell + 1)$ coefficients.

**Proof.** Consider the set of all subspace polynomials of subspaces in $C_g$ (Construction 3). Lemma 26 implies that these polynomials have zero coefficients for all monomials $x^j$ such that $g \nmid j$. Hence, they may be partitioned into $q^{n\ell}$ subsets according to their $\ell + 1$ top coefficients which correspond to monomials whose $q$-degree is divisible by $g$. According to the pigeonhole principle, there exists a subset of size at least $\binom{n/g}{r/g} q^n / q^{n\ell}$ whose polynomials agree on their top $g(\ell + 1)$ coefficients.

Notice that for $g = 1$, Theorem 23 reduces to the ordinary counting argument employed by [4] and [31]. In addition, the case where $n - r = g(\ell + 1) \geq r$, in which the polynomials in the set agree on all coefficients, is also trivial, since it merely implies the existence of a set of size one. Hence, this theorem is applicable only when $r > n/2$.

The notion of $q^g$-associativity, together with Lemma 24, allows us to construct an *explicit* large set of subspace polynomials. It will also be noted that in certain cases, this set of polynomials corresponds to the entire set $C_g$. The construction is based on the following lemma.

**Lemma 28** If $g, s,$ and $r$ are integers such that $gs \mid r$ and $n \triangleq r + gs$, then the polynomial $P(x) \triangleq \sum_{i=0}^{n/gs - 1} x^{igs}$ is a subspace polynomial with respect to $\mathbb{F}_{q^n}$.

**Proof.** Since $gs \mid r$, there exists an integer $\alpha$ such that $gs\alpha = r$, thus $n = gs(\alpha + 1)$ and $s \mid \frac{n}{g}$. It follows that

$$\frac{x^{n/g} - 1}{x^s - 1} = x^{\frac{n}{g} - s} + x^{\frac{n}{g} - 2s} + \ldots + 1,$$
and hence \((x^{n/g-s} + x^{n/g-2s} + \ldots + 1))((x^{n/g} - 1)).\) According to Lemma 27, the \(q^s\)-associates of these polynomials satisfy \(\sum_{i=0}^{n/g-1} x^{[ig]s}((x^n - x))\), and thus \(P\) is a subspace polynomial of an \(r\)-subspace in \(\mathbb{F}_{q^n}\) by Definition 7.

By Lemma 24 and Lemma 28, we have a large set of subspace polynomials whose coefficients may be given explicitly.

**Construction 4** If \(g, s,\) and \(r\) are integers such that \(gs|r\) and \(n \triangleq r + gs\), then

\[
Z \triangleq \left\{ \sum_{i=0}^{n/gs-1} \beta^{[r]-[ig]s} x^{[ig]s} \mid \beta \in B \right\}
\]

consists of \(\frac{q^n}{q^{gs}-1}\) subspace polynomials of subspaces in \(G_{q}(n, r)\), where \(B\) is any set of nonzero representatives of the orbit of \(\mathbb{F}_{q^{gs}}\).

**Proof.** Since \(n = r + gs\) and \(gs|r\), it follows that \(gs|n\), and thus \(\mathbb{F}_{q^{gs}}\) is a subfield of \(\mathbb{F}_{q^n}\). By Lemma 28, the polynomial \(P_V(x) = \sum_{i=0}^{n/gs-1} x^{[ig]s}\) is a subspace polynomial of some \(V \in G_{q}(n, r)\). Let \(B\) be any set of representatives of the orbit of \(\mathbb{F}_{q^{gs}}\), that is, a set consisting of a single nonzero element from each subspace in \(\{\alpha \mathbb{F}_{q^{gs}} \mid \alpha \in \mathbb{F}_{q^n}^*\}\). Since the size of the orbit of \(\mathbb{F}_{q^{gs}}\) is \(\frac{q^n}{q^{gs}-1}\), and since all subspaces in it intersect trivially [9, Section III], it follows that \(|B| = \frac{q^n}{q^{gs}-1}\). By Lemma 24 for all \(\beta \in B\) we have that \(P_{\beta V}(x) \in Z\). We are left to show that if \(\beta_1, \beta_2 \in B\), then \(\beta_1 V \neq \beta_2 V\).

Assume for contradiction that there exists \(\beta_1, \beta_2 \in B\) such that \(\beta_1 V = \beta_2 V\). It follows that \(P_{\beta_1 V}(x) = P_{\beta_2 V}(x)\), and Lemma 24 implies that the coefficients of \(x\) are equal, that is, \(\beta_1^{[n-gs]-1} = \beta_2^{[n-gs]-1}\). Therefore, since every \(\alpha \in \mathbb{F}_{q^n}\) satisfies \(\alpha^{q^n} = \alpha\), we have that

\[
\left(\beta_1^{[n-gs]-1}\right)^{-q^{gs}} = \left(\beta_2^{[n-gs]-1}\right)^{-q^{gs}}
\]

\[
\beta_1^{q^{gs}} - \beta_2^{q^{gs}} = \beta_2^{q^{gs}} - \beta_2^{q^{gs}}
\]

\[
\beta_1^{q^{gs}} = \beta_2^{q^{gs}}
\]

\[
\left(\frac{\beta_1}{\beta_2}\right)^{q^{gs} - 1} = 1.
\]

It is widely known (e.g., [18, Theorem 3.20, p. 91]) that the subspace polynomial of \(\mathbb{F}_{q^{gs}}\) is \(x^{q^{gs}} - x\), which implies that \(\beta_1 \beta_2^{-1} \in \mathbb{F}_{q^{gs}}\), and thus
\(\beta_1 \in \beta_2 F_{q^s}\). Since \(\beta_2 \in \beta_2 F_{q^s}\), it follows that \(\beta_1\) and \(\beta_2\) belong to the same cyclic shift \(\beta_2 F_{q^s}\), a contradiction. 

Notice that the set \(B\) of representatives of \(F_{q^s}\) (see Construction 4) may easily be found. For example, if \(\gamma\) is a primitive element of \(F_{q^n}\), since the set \(\{0\} \cup \{\gamma^{i(q^n-1)/(q^s-1)}\}_{i=0}^{q^n-2}\) is \(F_{q^s}\), it follows that a possible set of representatives of the orbit of \(F_{q^s}\) is

\[
B \triangleq \left\{ \gamma^i \mid 0 \leq i \leq \frac{q^n-1}{q^s-1} - 1 \right\}.
\]

**Remark 7** For \(s = 1\), the set \(Z\) from Construction 4 consists of all subspace polynomials of subspaces in \(G_q\) (see Construction 3). This is since the number of cyclic shifts of \(F_{q^s}\) is \(q^n - 1\) and the size of \(G_q\) is \(\left[\frac{n}{r}\right] = \left[\frac{n}{g}\right] = q^n - 1\).

In Section 5.4, we consider subspace polynomials over \(F_{q^m}\) as polynomials over an extension field \(F_{q^m}\) of \(F_{q^n}\). In order to use the above claims over \(F_{q^m}\), the following formal lemma is required. The proof of this lemma is an immediate corollary of the existence of an injective homomorphism \(\phi : F_{q^n} \to F_{q^m}\).

**Lemma 29** Let \(P_V(x) = x^r + \sum_{j=0}^{r-1} v_j x^j\) and \(P_U(x) = x^r + \sum_{j=0}^{r-1} u_j x^j\) be two subspace polynomials of subspaces in \(G_q(n, r)\), and let \(F_{q^m}\) be an extension field of \(F_{q^n}\). If we consider \(P_V, P_U\) as polynomials \(P_V', P_U'\) over \(F_{q^m}\), i.e.,

\[
P_V'(x) = x^r + \sum_{j=0}^{r-1} v'_j x^j
\]

\[
P_U'(x) = x^r + \sum_{j=0}^{r-1} u'_j x^j
\]

where the coefficients are in \(F_{q^m}\), then for all \(j \in \{0, \ldots, r - 1\}\), \(v_j = u_j\) if and only if \(v'_j = u'_j\). Furthermore, the polynomials \(P_V', P_U'\) are subspace polynomials in \(G_q(m, r)\).

Notice that generalizing Lemma 29 to the case where \(F_{q^m}\) is not an extension field of \(F_{q^n}\), i.e. \(U\) and \(V\) are subspaces in \(F_{q^m}\) which are contained in a
subspace of dimension \( n \), is not clear. However, such a generalization is necessary to use our techniques to bound the list size for any \( m \geq n \).

### 5.4 Improved Bounds on List Decodability of Gabidulin Codes

We begin by formally defining Gabidulin codes, which are rank-metric codes that attain a Singleton-like bound. Any rank-metric code over \( \mathbb{F}_{q^m} \) of length \( n \), minimum rank distance \( d \), and size \( M \) satisfies \( M \leq q^{m(n-d+1)} \) [6, 27]. For a linear rank-metric code of dimension \( k \), this bound implies that \( d \leq n - k + 1 \). Codes which attain this bound are called maximum rank distance (MRD) codes. It can be shown that Gabidulin codes, defined below, are linear MRD codes, attaining \( d = n - k + 1 \).

**Definition 9** [10] A linear Gabidulin code \( \text{Gab}[n, k] \) over \( \mathbb{F}_{q^m} \), length \( n \leq m \), and dimension \( k \leq n \) is the set

\[
\text{Gab}[n, k] \triangleq \{ (P(\alpha_1), \ldots, P(\alpha_n)) \mid \deg_q P < k \},
\]

where \( P \) traverses all \( q \)-degree restricted linearized polynomials, and \( \alpha_1, \ldots, \alpha_n \) are some fixed elements of \( \mathbb{F}_{q^m} \) which are linearly independent over \( \mathbb{F}_q \).

In [31] it was shown that large sets of subspace polynomials that agree on many top coefficients may be used to show the limits of list decoding of Gabidulin codes. For the lack of knowledge about the structure of the coefficients of subspace polynomials, a counting argument was later applied to show the existence of such a set. The resulting bound on list decoding of Gabidulin codes is cited below. In what follows, for \( w \in \mathbb{F}_{q^m}^n \) and \( \tau \in \mathbb{N} \), let \( B_\tau(w) \triangleq \{ c \mid \text{rank}(w - c) \leq \tau \} \), that is, a ball of radius \( \tau \) centered at \( w \).

**Theorem 24** [31, Theorem 1] Consider the code \( \text{Gab}[n, k] \) over \( \mathbb{F}_{q^m} \), with \( d = n - k + 1 \). If \( \tau < d \), then there exists a word \( w \in \mathbb{F}_{q^m}^n \) such that

\[
|\text{Gab}[n, k] \cap B_\tau(w)| \geq \frac{\binom{n}{n-\tau} q^{(q^m)^{n-\tau-k}}}{n}
\]

As a result, the following bound is achieved.
Corollary 15 \[31\], Section III] The code \(\text{Gab}[n,k]\) over \(\mathbb{F}_{q^m}\), with \(d = n - k + 1\) cannot be list decoded efficiently for any list decoding radius 
\[
\tau \geq \frac{m + n}{2} - \sqrt{\frac{(m+n)^2}{4} - m(d-\varepsilon)},
\]
for any fixed \(0 \leq \varepsilon < 1\).

For \(n = m\), this bound simplifies to
\[
\tau \geq n - \sqrt{n(n-d+\varepsilon)},
\]
which may be seen as the rank-metric equivalent of the Johnson radius [13], and for \(\varepsilon = 0\) it is equal to the Hamming-metric Johnson radius.

By Lemma [26] in certain cases there exists a large set of subspace polynomials with a unique coefficient structure. Restricting the counting argument used in the proof of Theorem [24] to the set \(C_g\) (Theorem [23]) provides a bound which may outperform Corollary 15. The proof of the following theorem is illustrated in Fig. 5.1, and its consequences are discussed in the sequel.

Theorem 25 For integers \(k \leq n \leq m\) such that \(n\) divides \(m\), let \(\text{Gab}[n,k]\) be a linear Gabidulin code over \(\mathbb{F}_{q^m}\), with \(d = n - k + 1\) and evaluation points \(\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}\). Let \(\tau, g\) be integers such that \(\left\lfloor \frac{d-1}{2} \right\rfloor + 1 \leq \tau \leq d - 1\), \(g \geq 2\), and \(g | \gcd(n - \tau, n)\). If \(\ell\) is the unique integer such that \(n = n - \tau + g(\ell + 1)\) (and thus, \(\tau = g(\ell + 1)\)), then there exists a word \(c_R \in \mathbb{F}_{q^m}^n \setminus \text{Gab}[n,k]\) such that
\[
|\text{Gab}[n,k] \cap B_\tau(c_R)| \geq \frac{\left\lfloor \frac{n}{g(\ell + 1)} \right\rfloor q^g}{q^{\ell \tau}}.
\]

**Proof.** According to Theorem [23], there exists a set \(P\) of \(\left\lfloor \frac{n}{g(\ell + 1)} \right\rfloor q^g/q^{n\ell}\) subspace polynomials of subspaces in \(G_q(n, n - \tau)\), that agree on their topmost \(\tau = g(\ell + 1)\) coefficients. The coefficients of these polynomials are in the field \(\mathbb{F}_{q^n}\). Since \(n|m\), we have that \(\mathbb{F}_{q^n}\) is a subfield of \(\mathbb{F}_{q^m}\), and thus these coefficients may be considered as elements of \(\mathbb{F}_{q^m}\). Recall that according to Lemma [29] these polynomials agree on their topmost \(\tau\) coefficients also when considered as polynomials over \(\mathbb{F}_{q^m}\). Let \(R\) be any linearized polynomial over \(\mathbb{F}_{q^m}\) of \(q\)-degree \(n - \tau\) that has these top coefficients, and
Figure 5.1: An illustration of the proof of Theorem 25. The proof of Theorem 26 is similar. The ball around $c_R$ of radius $\tau$ contains the words $c_R - P_i$ for $P_i \in \mathcal{P}$, where $|\mathcal{P}| = \left\lfloor \frac{n}{g(n-\tau)} \right\rfloor / q^{g\ell}$.

let $c_R \in \mathbb{F}_{q^m}^n$ be the word resulting from the evaluation of $R$ at $\alpha_1, \ldots, \alpha_n$. Similarly, for $P \in \mathcal{P}$ let $c_{R-P} \in \mathbb{F}_{q^m}^n$ be the word corresponding to the evaluation of $R - P$ at $\alpha_1, \ldots, \alpha_n$.

Since $\deg_q(R - P) \leq n - \tau - g(\ell + 1)$ and $\tau = g(\ell + 1) > \frac{d-1}{2} = \frac{n-k}{2}$ it follows that $2\tau = \tau + g(\ell + 1) > n - k$, and hence,

$$k > n - \tau - g(\ell + 1) \geq \deg_q(R - P).$$

Therefore, the word $c_{R-P}$ is a codeword of $\text{Gab}[n,k]$ for all $P \in \mathcal{P}$. In addition, since $\tau \leq d - 1$ it follows that $\deg_q R = n - \tau \geq n - d + 1 = k$, and hence $c_R \notin \text{Gab}[n,k]$. 

79
Since every linearized polynomial can be viewed as an \( \mathbb{F}_q \)-linear mapping (see Section 5.1), it follows that for every \( P \in \mathcal{P} \),

\[
\text{rank}(cR - cR - P) = \text{rank}(P(\alpha_1), \ldots, P(\alpha_n)) \\
\leq \dim \text{Im}(P) = n - \dim \ker(P) \\
= n - (n - \tau) = \tau.
\]

Thus, the set \( \{cR - P\}_{P \in \mathcal{P}} \subseteq \text{Gab}[n, k] \) is a set of size \( \left[ \frac{n/g}{(n-\tau)/g} \right] q^{n\ell} \), which is contained in a ball of radius \( \tau \) around the word \( cR \).

Notice that the restriction on the parameter \( r \), mentioned after the proof of Theorem 23, implies the necessary condition \( r = n - \tau > n/2 \), and hence \( \tau < n/2 \). However, this limitation becomes trivial when discussing \( \tau \) which is approximately the unique decoding radius \( d/2 \), since \( d \leq n \).

A simple analysis of (5.1) shows that

\[
|\text{Gab}[n, k] \cap B_\tau(cR)| \geq \left[ \frac{n/g}{(n-\tau)/g} \right] q^{n\ell} \\
\geq \left( q^0 \right)^{\frac{n-\tau}{g} - \frac{n - \tau}{g}} q^{n\ell} \\
= q^{(n-\tau)\frac{\ell}{g} - n\ell} = q^{\frac{n\tau}{g} - \frac{\ell^2}{g} - n\ell} \\
= q^{n(\ell+1) - g(\ell+1)^2 - n\ell} \\
= q^{n-g(\ell+1)^2} = q^{n-\tau(\ell+1)},
\]

and hence, this bound results in a list of exponential size whenever \( g(\ell+1)^2 < c \cdot n \) for \( c \in (0, 1) \), or alternatively, when \( \tau < \frac{cn}{\ell+1} \).

The following examples provide infinite sets of Gabidulin codes, with rates from \( \frac{1}{2} \) to 1, that cannot be list decoded efficiently at all according to the bound from Theorem 25. This result strictly outperforms the bound from Corollary 15 and provides an answer to an open problem by [7, 15, Section 6], and [31, Section V], that is, there exist Gabidulin codes that cannot be efficiently list decoded beyond the unique decoding radius.

**Example 3** Let \( n \) be an integer power of 2, and let \( 1 \leq i \leq \log n - 2 \). For any integer \( m \) such that \( n|m \), consider the code \( \text{Gab}[n, (1 - \frac{1}{2^i})n + 2] \) over
\( \mathbb{F}_{q^m} \), and let \( \tau \) be the smallest possible list decoding radius, that is,

\[
\tau \triangleq \left\lfloor \frac{d-1}{2} \right\rfloor + 1 = \left\lfloor \frac{n}{2^{i+1}} - \frac{2}{2} \right\rfloor + 1 = \frac{n}{2^{i+1}} + 1.
\]

Let \( g \triangleq \frac{n}{2^{i+1}} = \tau \), and notice that \( g \geq 2 \). To see that \( g \) divides \( \gcd(n, n-\tau) \), notice that since \( n \) is an integer power of 2, it follows that \( \tau \) divides \( n \), and thus \( g \) divides \( n \). In addition, we have that \( \tau(2^{i+1} - 1) = n - \tau \), thus \( \tau \) divides \( (n - \tau) \) and \( g \) divides \( (n - \tau) \).

Therefore, in Theorem 25 we may choose \( g = \frac{n}{2^{i+1}} \), \( \ell = 0 \), and get that there exists a word \( c_R \in \mathbb{F}_{q^m}^n \) with \( q \left( 1 - \frac{2}{2^{i+1}} \right) n \) codewords in a ball of radius \( \tau \) around it. Since \( \tau \) is larger than the unique decoding radius by one, this code cannot be efficiently list decoded at all. A detailed comparison between this bound and [31] appears in .

**Example 4** Let \( g, \alpha_n, \) and \( \alpha_\tau \) be positive integers such that \( \alpha_n \geq \alpha_\tau^2 + 1 \).

For \( n = \alpha_n g, \tau = \alpha_\tau g \), and any integer \( m \) such that \( n \mid m \), consider the code \( \text{Gab}[n, n-2\tau+1] \) over \( \mathbb{F}_{q^m} \), whose minimum distance is \( d = 2\tau \), and whose rate is

\[
\frac{n - 2\tau + 1}{n} = 1 - \frac{2\alpha_\tau}{\alpha_n} + \frac{1}{n}.
\]

According to Theorem 25, there exists a word \( c_R \) having

\[
\frac{\left\lfloor \frac{n}{(n-\tau)/g} \right\rfloor}{q^{n\ell}} q^\alpha_n \tag{5.2}
\]

codewords in radius \( \tau \) around it, where \( \ell = \tau/g - 1 = \alpha_\tau - 1 \). Simplifying this expression, we have that

\[
\frac{\left\lfloor \frac{n}{(n-\tau)/g} \right\rfloor}{q^{n\ell}} = \frac{\left\lfloor \frac{\alpha_n}{\alpha_n-\alpha_\tau} \right\rfloor}{q^{n(\alpha_\tau-1)}} \geq \frac{(q^\alpha)^{(\alpha_n-\alpha_\tau)}^{\alpha_\tau}}{q^{n(\alpha_\tau-1)}} = q^{n-\tau\alpha_\tau} = q^{(\alpha_n-\alpha_\tau^2)g}.
\]

If \( \alpha_\tau \) and \( \alpha_n \) are constants then \( g = \Omega(n) \) and \( q^{(\alpha_n-\alpha_\tau^2)g} = q^{\Omega(n)} \), which implies that the list size is exponential in the code length. Since \( \tau < n/2 \), as mentioned after Theorem 25, it follows that \( \alpha_n > 2\alpha_\tau \), and thus we have
the following two interesting families of codes.

1. For $\alpha_n = 3$ and $\alpha_\tau = 1$ we have the code $\text{Gab}[3g, g + 1]$ over any field $\mathbb{F}_{q^m}$ such that $3g|m$. The rate of this code is $\frac{1}{3} + \frac{1}{n}$, and its minimum distance is $2g$. For the radius $\tau = g$, there exists a word $c_R$ with at least $q^{2g} = q^{\Omega(n)}$ codewords around it, and hence this code cannot be list decoded efficiently at all.

2. For $\alpha_n = 5$ and $\alpha_\tau = 2$ we have the code $\text{Gab}[5g, g + 1]$ over any field $\mathbb{F}_{q^m}$ such that $5g|m$. The rate of this code is $\frac{1}{5} + \frac{1}{n}$, and its minimum distance is $4g$. For the radius $\tau = 2g$, there exists a word $c_R$ with at least $q^g = q^{\Omega(n)}$ codewords around it, and hence this code cannot be list decoded efficiently at all.

Clearly, this strategy can be used to construct examples of families with larger code rates, but $\frac{1}{3} + \frac{1}{n}$ is the smallest one. This may be seen by considering all integers $\alpha_\tau$ and $\alpha_n$ which comply with the above constraints. That is, for $\alpha_\tau = 1$ and $\alpha_n \geq 4$, the rate is at least $\frac{1}{2} + \frac{1}{n}$, for $\alpha_\tau = 2$ and $\alpha_n \geq 6$ the rate is at least $\frac{1}{3} + \frac{1}{n}$, and for any $\alpha_\tau \geq 3$ and any $\alpha_n \geq \alpha_\tau^2 + 1$ the rate is at least $\frac{1}{3} + \frac{1}{n}$.

In the following, we present a simple algorithmic way of constructing many dense sets of Gabidulin codewords. These sets also show that the corresponding Gabidulin codes cannot be efficiently list decoded beyond the unique decoding radius. In addition, we have that for certain Gabidulin codes, dense sets of codewords abound and may easily be computed explicitly.

**Theorem 26** Let $g, s, n,$ and $m$ be integers such that $g \geq 2$, $gs|n$, and $n|m$. Let $\text{Gab}[n, n - 2gs + 1]$ be a linear Gabidulin code over $\mathbb{F}_{q^m}$, with $d = 2gs$ and evaluation points $\alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m}$. If $\tau \triangleq \lfloor \frac{d-1}{2} \rfloor + 1 = gs$, then there exists an (explicitly defined) word $c_R \in \mathbb{F}_{q^m}^n \setminus \text{Gab}[n, n - 2gs + 1]$ such that

$$|\text{Gab}[n, n - 2gs + 1] \cap B_\tau(c_R)| \geq \frac{q^n - 1}{q^{gs} - 1}.$$ 

In particular, if $R$ is the polynomial whose evaluation in $\alpha_1, \ldots, \alpha_n$ yields $c_R$, then $\frac{q^n - 1}{q^{gs} - 1}$ of the codewords in $B_\tau(c_R)$ are given by the evaluations of $\{R - P\}_{P \in \mathbb{Z}}$ (see Construction 4) in $\alpha_1, \ldots, \alpha_n$. 

82
Proof. Since $gs | n - gs$, by setting $r = n - gs$ it follows from Construction 4 that the set $\mathcal{Z}$ is a set of subspace polynomials of subspaces in $G_q(n, n - gs)$, whose size is $q^{n-1}/q^{gs-1}$. Since $n | m$, we have that $\mathbb{F}_{q^n}$ is a subfield of $\mathbb{F}_{q^m}$, and therefore the polynomials in $\mathcal{Z}$ may be considered as polynomials over $\mathbb{F}_{q^m}$ as well. According to Construction 4 and Lemma 29, the polynomials in $\mathcal{Z}$ agree on their topmost $gs$ coefficients $(1, 0, \ldots, 0)$, even when considered as polynomials over $\mathbb{F}_{q^m}$.

Let $R$ be any linearized polynomial of $q$-degree $n - gs$ whose top $gs$ coefficients are $(1, 0, \ldots, 0)$, and let $c_R \in \mathbb{F}_{q^m}^n$ be the word resulting from the evaluation of $R$ at $\alpha_1, \ldots, \alpha_n$. For each $P \in \mathcal{Z}$ let $c_{R-P} \in \mathbb{F}_{q^m}^n$ be the word corresponding to the evaluation of $R - P$ at $\alpha_1, \ldots, \alpha_n$. For all $P \in \mathcal{Z}$ we have that $\deg_q(R - P) \leq n - 2gs < n - 2gs + 1$, and thus $c_{R-P} \in \text{Gab}[n, n - 2gs + 1]$. In addition, $\deg_q R = n - gs$, and thus $c_R \notin \text{Gab}[n, n - 2gs + 1]$.

As in the proof of Theorem 25 for all $P \in \mathcal{Z}$ we have that $\text{rank}(c_R - c_{R-P}) \leq n - \dim \ker(P) = gs$. Therefore, the set $\{c_{R-P} \mid P \in \mathcal{Z}\}$ is a set of $q^{n-1}/q^{gs-1}$ codewords in $\text{Gab}[n, n - 2gs + 1]$, all of which are of distance at most $\tau = gs$ from $c_R$. 

Notice that each code in the family of codes mentioned in Theorem 26 satisfies $d = 2gs$, and hence the unique decoding radius is $\lfloor \frac{d-1}{2} \rfloor = gs - 1$. Furthermore, since $gs | n$, it follows that $gs \leq \frac{n}{2}$, and thus the word $c_R$ has $\Omega(q^{n/2})$ codewords in a ball of radius $\tau = \lfloor \frac{d-1}{2} \rfloor + 1$ around it. Hence, this family of Gabidulin codes cannot be list decoded efficiently at all.

It is an interesting question if our results can be used to derive a lower bound on the number of words that have an exponentially-sized list of codewords around themselves. If it can be proved that there are just a few just words, we might be able to remove a few codewords of the Gabidulin code to obtain a list decodable code of slightly smaller rate. The code constructed in [15] seems to be such a list decodable code.

Further, for folded Gabidulin codes such a subcode might be easy to find. The results from [1] show that the average list size of folded Gabidulin codes is quite small, indicating that there are only a few words with an exponentially-sized list around them.

Finally, the results in this section can be used to prove bounds for punctured Gabidulin codes, which are obtained by removing coordinates from the original code. Puncturing a $\text{Gab}[n, k]$ code by $s < n - k + 1$ positions...
yields a Gab[n\ −\ s, k] code. We can therefore provide lower bounds on list decoding of Gabidulin codes where n does not divide m.

Lemma 30 Let C be a Gab[n, k] code over \( \mathbb{F}_{q^m} \) with minimum distance \( d \triangleq n-k+1 \), let \( s \) be an integer such that \( s < d \), and let \( C_s \) be a Gab[n−s, k] code which results from \( C \) by \( s \) puncturing operations, whose minimum distance is \( d' \triangleq n-s-k+1 \). If \( C \) cannot be list decoded efficiently at all, i.e., there exists a word \( w \in \mathbb{F}_{q^m}^n \) such that
\[
|C \cap B_\tau(w)| \geq q^{\Omega(n)}
\]

where \( \tau \triangleq \lfloor \frac{d-1}{2} \rfloor + 1 \), then \( C_s \) cannot be list decoded efficiently for any radius \( \tau' + \frac{s'}{2} \), where \( \tau' = \lfloor \frac{d'-1}{2} \rfloor + 1 \), and
1. If \( s \) is even, then \( s' = \frac{s}{2} \).
2. If \( s \) is odd and \( n-k \) is even, then \( s' = \frac{s}{2} + \frac{1}{2} \).
3. If \( s \) and \( n-k \) are both odd, then \( s' = \frac{s}{2} - \frac{1}{2} \).

Proof. Since puncturing may only reduce the distance between any two given words, and since any two codewords in \( C \) cannot coincide by puncturing \( s < d \) coordinates, it follows that
\[
|C_s \cap B_\tau(w')| \geq q^{\Omega(n)},
\]

where \( w' \in \mathbb{F}_{q^m}^{n-s} \) is the result of puncturing \( w \). Hence, \( C_s \) cannot be list decoded efficiently beyond the radius \( \tau \). Table 5.1 presents the values of \( \tau \) as a function of \( \tau' \) and \( s \), from which the claim follows.

Since the addition to the unique decoding radius \( \tau' \) of Gab[n−s, k] in Lemma 30 is usually nonzero, it is not clear if those punctured codes indeed cannot be list decoded efficiently at any radius. However, for the special case where \( s = 1 \) and \( n-k \) is odd, we obtain the following corollary.

Corollary 16 For integers 0 < k < n such that \( n-k \) is odd, if Gab[n, k] cannot be list decoded efficiently at all, i.e., there exist a word \( w \in \mathbb{F}_{q^m}^n \) such that
\[
|C \cap B_\tau(w)| \geq q^{\Omega(n)}
\]

where \( \tau \triangleq \lfloor \frac{d-1}{2} \rfloor + 1 \), then the punctured code Gab[n−1, k] cannot be list decoded efficiently at all.
\[ \tau = \left\lfloor \frac{n-k}{2} \right\rfloor + 1 \quad \tau' = \left\lfloor \frac{n-k-s}{2} \right\rfloor + 1 \quad \text{Resulting radius} \]

<table>
<thead>
<tr>
<th>Condition</th>
<th>( n-k ) and ( s )</th>
<th>( n-k ) is odd and ( s ) is even</th>
<th>( n-k ) is even and ( s ) is odd</th>
<th>( n-k ) and ( s ) are both odd</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n-k ) and ( s ) are both even.</td>
<td>( \frac{n-k}{2} + 1 )</td>
<td>( \frac{n-k-s}{2} + 1 )</td>
<td>( \tau = \tau' + \frac{s}{2} )</td>
<td>( \tau = \tau' + \frac{s}{2} )</td>
</tr>
<tr>
<td>( n-k ) is odd and ( s ) is even.</td>
<td>( \frac{n-k-1}{2} + 1 )</td>
<td>( \frac{n-k-1-s}{2} + 1 )</td>
<td>( \tau = \tau' + \frac{s}{2} )</td>
<td>( \tau = \tau' + \frac{s}{2} )</td>
</tr>
<tr>
<td>( n-k ) is even and ( s ) is odd.</td>
<td>( \frac{n-k}{2} + 1 )</td>
<td>( \frac{n-k-s+1}{2} + 1 )</td>
<td>( \tau = \tau' + \frac{s+1}{2} )</td>
<td>( \tau = \tau' + \frac{s+1}{2} )</td>
</tr>
<tr>
<td>( n-k ) and ( s ) are both odd.</td>
<td>( \frac{n-k-1}{2} + 1 )</td>
<td>( \frac{n-k-1-s-1}{2} + 1 )</td>
<td>( \tau = \tau' + \frac{s-1}{2} )</td>
<td>( \tau = \tau' + \frac{s-1}{2} )</td>
</tr>
</tbody>
</table>

Table 5.1: The resulting radius in Lemma 30. If Gab\([n, k]\) cannot be list decoded efficiently for the radius \( \tau \), then the punctured code Gab\([n-s, k]\), \( s < n-k+1 \), cannot be list decoded efficiently for this radius as well. The rightmost column provides \( \tau \) as a function of \( \tau' \) and \( s \), where the unique decoding radius of Gab\([n-s, k]\) is \( \tau' - 1 \). The given values for \( \tau' \) are simple calculations which follow from \( n-k-s \) being either even or odd.

Although Corollary 16 does not provide a drastic improvement in the variety of codes to which our bounds apply, it does imply the important observation that the divisibility constraints between \( n \) and \( m \) in Theorem 25 and Theorem 26 are not necessary. In addition, one may obtain infinite examples of Corollary 16 by puncturing either of the codes Gab\([3g, g+1]\) and Gab\([5g, g+1]\) from Example 4 and thus obtain that the codes Gab\([3g-1, g+1]\) and Gab\([5g-1, g+1]\) cannot be list decoded efficiently at all.

5.5 Bounds for Constant-Dimension Subspace Codes

In this section, we state new bounds on list decoding lifted Gabidulin codes (see [30]), which are a class of almost-optimal constant dimension subspace codes. Lifted Gabidulin codes are of special interest since, in contrast to many other subspace code constructions, they can be efficiently decoded (see [30]) while only losing a relatively small number of codewords compared to other subspace code constructions. These bounds are a direct
consequence of our bounds for list decoding Gabidulin codes (Theorem 25 and Theorem 26).

Throughout this section, the quadruple \((n, M_s, d_s, r)\) denotes a constant dimension subspace code in the Grassmannian \(G_q(n, r)\) of cardinality \(M_s\) and minimum subspace distance \(d_s\). Further, \(\langle A \rangle\) denotes the subspace spanned by the rows of a matrix \(A\). The lifting is a map which is applied to a single matrix or a set of matrices and is defined as follows.

**Definition 10** Consider the mapping

\[
\mathcal{I} : \mathbb{F}_q^{n \times m} \to G_q(n, n + m)
\]

\[
X \mapsto \langle [I_n \ X] \rangle,
\]

where \(I_n\) denotes the \(n \times n\) identity matrix. The subspace \(\mathcal{I}(X) = \langle [I_n \ X] \rangle\) is called lifting of the matrix \(X\). If we apply this map on all codewords of a code \(C\) (in matrix representation), then the subspace code \(\mathcal{I}(C)\) is called lifting of the code \(C\).

The properties of a lifted code were studied by Silva, Kschischang and Kötter and are summarized in the following two lemmas.

**Lemma 31** Let \(X, Y \in \mathbb{F}_q^{n \times m}\) and let \(\mathcal{I}(X), \mathcal{I}(Y) \in G_q(n + m, n)\) be as in Definition 10. Then,

\[
d_s(\mathcal{I}(X), \mathcal{I}(Y)) = 2 \cdot d_R(X, Y).
\]

**Proof.**

\[
d_s(\mathcal{I}(X), \mathcal{I}(Y)) = 2 \dim(\mathcal{I}(X) + \mathcal{I}(Y))
\]

\[
- \dim(\mathcal{I}(X)) - \dim(\mathcal{I}(Y))
\]

\[
= 2 \ \text{rank} \begin{pmatrix} I_n & X \\ I_n & Y \end{pmatrix} - 2n
\]

\[
= 2 \ \text{rank} \begin{pmatrix} I_n & X \\ 0 & Y - X \end{pmatrix} - 2n
\]

\[
= 2 \left[ \text{rank}(I_n) + \text{rank}(X - Y) \right] - 2n
\]

\[
= 2 \ \text{rank}(X - Y) = 2d_R(X, Y).
\]
The following lemma directly follows from Lemma 31:

Lemma 32 [30] Let \( C \) be a rank-metric code over \( \mathbb{F}_{q^m} \) of length \( n \leq m \), minimum rank distance \( d_R \) and cardinality \( M_R \), whose codewords are represented as \( m \times n \) matrices over \( \mathbb{F}_q \). Then, the lifting of the transposed codewords, i.e.,

\[
\mathcal{I}(C^T) \triangleq \{ \langle [I_n \ C^T] \rangle \mid C \in C \}
\]

is an \((n + m, M_s, d_s = 2d_R)\) constant dimension subspace code.

Hence, the lifting of the transpose of a \( \text{Gab}[n,k] \) code over \( \mathbb{F}_{q^m} \) with \( n \leq m \), minimum rank distance \( d = n - k + 1 \) and cardinality \( M_R = q^{mk} \) results in an \((n + m, q^{mk}, 2d, n)_q\) constant dimension subspace code in the Grassmannian \( G_q(n + m, m) \).

So far, the only known bound to list decoding subspace codes was given in [26] and is based on the results for Gabidulin codes from [31]. The following theorem summarizes the result from [26].

Theorem 27 [26, Theorem 37] Let \( C \) be a linear \( \text{Gab}[n,k] \) Gabidulin code over \( \mathbb{F}_{q^m} \) of length \( n \leq m \), \( d = n - k + 1 \), evaluation points \( \alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m} \), and let \( \tau \) be an integer such that \( \lfloor \tau/2 \rfloor < d \). Denote by \( \mathcal{I}(C^T) \) the \((n + m, q^{mk}, 2d, n)_q\) subspace code from the lifting of the code \( C \) as in Definition 10. Then, there is a subspace \( \langle R \rangle \) such that

\[
| C \cap B_\tau^s(\langle R \rangle) | \geq \frac{\binom{n}{\lfloor \tau/2 \rfloor}_q}{q^{m(n-k-(\tau/2))}}.
\]

Let \( B_\tau^s(\langle W \rangle) \triangleq \{ \langle V \rangle \mid d_s(\langle W \rangle, \langle V \rangle) \leq \tau \} \) denote a ball of radius \( \tau \) centered at \( \langle W \rangle \) in the subspace distance. With Lemma 31, we obtain the following relation between a rank-metric code \( C \) and its lifted subspace code \( \mathcal{I}(C^T) \):

\[
| C \cap B_\tau(c_R) | \leq | \mathcal{I}(C^T) \cap B_\tau^s(\mathcal{I}(c_R^T)) |. \quad (5.3)
\]

This relation and Theorem 25 provide the following theorem on the list size of lifted Gabidulin codes.
Theorem 28 Let \( C \) be a linear Gab\([n,k]\) Gabidulin code over \( \mathbb{F}_{q^m} \) with length \( n \mid m \), \( d = n - k + 1 \), evaluation points \( \alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m} \), and let \( \tau, g \) be integers such that \( \lfloor \frac{d - 1}{2} \rfloor + 1 \leq \lfloor \frac{d}{2} \rfloor \leq d - 1 \), \( g \geq 2 \), and \( g \mid \gcd(n - \lfloor \frac{d}{2} \rfloor, n) \). Let \( \ell \) be the unique integer such that \( n = n - \lfloor \frac{\tau}{2} \rfloor + g(\ell + 1) \) (and thus, \( \lfloor \frac{\tau}{2} \rfloor = g(\ell + 1) \)) and denote by \( \mathcal{I}(C^T) \) the \((n + m, q^{nk}, 2d, n)_q\) subspace code from the lifting of the code \( C \) as in Definition 10.

Then there exists a subspace \( \mathcal{I}(c^T_R) \in \mathcal{G}(n + m, n) \), where \( c_R \in \mathbb{F}_{q^m}^n \setminus \text{Gab}[n, k] \) such that

\[
|\mathcal{I}(C^T) \cap B_{\ell}(\mathcal{I}(c^T_R))| \geq \frac{\left\lfloor \frac{n}{g} \right\rfloor}{q^n} \geq \frac{q^n - 1}{q^{\frac{\tau}{2}} - 1}.
\]

**Proof.** The statement follows from (5.3) and Theorem 25. The floor operation for \( \lfloor \frac{\tau}{2} \rfloor \) is necessary since the subspace distance is an even number, see explanation of the proof of [26, Theorem 37].

Thus, this bound results in a list of exponential size for even \( \tau \) when \( \tau < \frac{2cn}{c+1} \) and for odd \( \tau \) when \( \tau < \frac{2cn}{c+1} + 1 \) for \( c \in (0, 1) \), which results for many cases in a better bound than the one from [26, Theorem 37]. Similarly, from Theorem 26 we obtain the following theorem.

Theorem 29 Let \( g, s, n, \) and \( m \) be integers such that \( g \geq 2 \), \( gs | n \), and \( n | m \). Let \( C \) be a linear Gab\([n,n - 2gs + 1]\) Gabidulin code over \( \mathbb{F}_{q^m} \), with \( d = 2gs \) and evaluation points \( \alpha_1, \ldots, \alpha_n \in \mathbb{F}_{q^m} \). Denote by \( \mathcal{I}(C^T) \) the \((n + m, q^{m(n - 2gs + 1)}, 2d, n)_q\) subspace code from the lifting of the code \( C \) as in Definition 10.

If \( \lfloor \frac{s}{2} \rfloor \triangleq \lfloor \frac{d - 1}{2} \rfloor + 1 = gs \), then there exists an (explicitly defined) subspace \( \mathcal{I}(c^T_R) \in \mathcal{G}(n + m, n) \), where \( c_R \in \mathbb{F}_{q^m}^n \setminus \text{Gab}[n, n - 2gs + 1] \), such that

\[
|\mathcal{I}(C^T) \cap B_{\ell}(\mathcal{I}(c^T_R))| \geq \frac{q^n - 1}{q^{gs} - 1} = \frac{q^n - 1}{q^{\lfloor \frac{\tau}{2} \rfloor} - 1}.
\]

The explicitly defined subspace follows directly from lifting the matrix representation of the explicit word of Theorem 26.

In [31], a non-linear rank-metric code was presented which cannot be list decoded efficiently at all. The lifting of this code obviously results in a subspace code with the same restrictions to list decoding as lifted Gabidulin
codes. However, lifted Gabidulin codes are of special interest for network coding and therefore, we have analyzed their list decoding capability in this section.

## 5.6 Conclusions and Future Work

We have improved the worst-case bound on the list decodability of Gabidulin codes in many cases. This was shown by using the structure of the subspace polynomials of a subset of $\mathcal{G}_q(n, r)$ for $n$ and $r$ that have a non-trivial gcd. In addition, we have presented such subspace polynomials explicitly, using the notions of cyclic shifts and $q$-associativity. Both of these results outperform the counting argument applied in [31], and provide examples of infinite families of Gabidulin codes that cannot be list decoded efficiently beyond the unique decoding radius. This resolves an open question by [7, 15], and [31] and reveals a significant difference between decoding Gabidulin and Reed–Solomon codes despite their similar code definitions.

The work of [31] ruled out the existence of an efficient algorithm for list decoding of Gabidulin codes beyond the Johnson radius. Our work rules out the existence of an efficient list decoding algorithm that applies for any Gabidulin code and any radius beyond half the minimum distance. However, this certainly does not rule out the existence of an efficient algorithm for list decoding of very large subcodes of Gabidulin codes or Gabidulin codes with lower rates, since our work requires the code parameters to satisfy some strict number-theoretic constraints, and our examples have rate at least $\frac{1}{5}$. For example, [15] provides a subcode of a Gabidulin code which can be list decoded efficiently.

We have also shown that identical results hold for lifted Gabidulin codes, which are an important class of nearly optimal subspace codes. Additional discussion about the inapplicability of our techniques to improve the known combinatorial bound on list decoding of Reed–Solomon codes appears in [29].

For future research, we would like to have similar bounds on Gabidulin codes in $\mathbb{F}_{q^m}$ for any case where $n$ does not necessarily divide $m$, a problem which seems to require generalizing Lemma 29 to the case $n \nmid m$. In addition, we would like to derive bounds for Gabidulin codes with rates less than $\frac{1}{5}$. 
Appendix A

In [4], limits for list decoding of Reed–Solomon codes were shown using techniques which highly resemble the ones in [31] and in this paper. The interested reader might conjecture that the improvement achieved here (see Theorem 25 and Theorem 26) for Gabidulin codes may also be attained for Reed–Solomon codes, for which list decoding related problems were very extensively studied. In what follows we briefly describe why such an improvement cannot be directly attained by our techniques. Adapting these techniques to Reed–Solomon codes remains an intriguing open problem. In the sequel, we briefly describe the methods and results of [4].

Following the notations in [4], a Reed–Solomon code RS[q^n, q^u] of length q^n and dimension q^u is a subset of F_q^n such that

\[
\text{RS}[q^n, q^u] = \left\{ (p(\alpha_1), \ldots, p(\alpha_{q^n})) \mid p : F_q^n \to F_q^n \text{ is a polynomial with } \deg(p) < q^u \right\},
\]

where \( \{\alpha_i\}_{i=1}^{q^n} \) are all elements of F_q^n. Notice that Reed–Solomon codes may be defined as the evaluation of polynomials in any number of elements in the field. However, we consider this definition for convenience. Notice also that any word \( w \in F_q^n \) may be regarded as a polynomial over F_q^n, and any word \( c \in \text{RS}[q^n, q^u] \) may be regarded as a polynomial over F_q^n of bounded degree.

**Definition 11** [4, Definition 3.3] A family of polynomials \( \mathcal{P} \subseteq F_q^n[x] \) is said to be an \((a, s)\)-family if

1. Each polynomial in \( \mathcal{P} \) has at least \( a \) roots in \( F_q^n \).
2. There is a polynomial \( P^* \) such that for all \( P \in \mathcal{P} \), \( P^* - P \) has degree at most \( s \). We refer to \( P^* \) as a pivot of the family.

**Lemma 33** [4, Proposition 3.5] Let \( a, s \) and \( \ell \) be positive integers. Then, the following are equivalent.

1. There is a word \( w : F_q^n \to F_q^n \) and \( \ell \) polynomials \( P_1, \ldots, P_\ell \) of degree at most \( s \) such that for \( i = 1, 2, \ldots, \ell \), \( P_i \) and \( w \) agree on at least \( a \) points of \( F_q^n \).
2. There is an \((a, s)\)-family of size \(\ell\) of polynomials, whose pivot is the unique polynomial \(P_w\) that agrees with the word \(w\) on all elements in \(\mathbb{F}_{q^n}\).

The polynomial \(P_w\) corresponds to the “problematic” word, that is, the word that has exponentially many codewords in a small radius around it. The polynomials \(P_1, \ldots, P_\ell\), having a low degree, are the codewords surrounding \(P_w\). It is readily verified that the polynomials \(P_1, \ldots, P_\ell\) are inside a ball of small radius centered at \(P_w\) if and only if the polynomials \(\{P_w - P_i\}_{i=1}^s\) have multiple roots in \(\mathbb{F}_{q^n}\). As subspace polynomials have many roots over \(\mathbb{F}_{q^n}\), they are good candidates for playing the role of the polynomials \(\{P_w - P_i\}_{i=1}^s\). This intuition is formalized as follows.

**Lemma 34** [4] If \(S \subseteq \mathcal{G}_q(n, r)\) is a set of subspaces whose corresponding subspace polynomials have identical \(r-t\) top coefficients for some integer \(t < r\), then the set of subspace polynomials of \(S\) forms a \((q^r, q^t)\)-family.

**Proof.** Let \(W\) be the set of subspace polynomials of the subspace in the set \(S\). Since every polynomial in \(W\) is a subspace polynomial, it has exactly \(q^r\) roots in \(\mathbb{F}_{q^n}\). If \(P_w\) is the linearized polynomial consisting of the \(r-t\) mutual top coefficients of the polynomials in \(W\), then \(\deg(P_w - P_i) \leq q^t\) for all \(P_i \in W\). \[\blacksquare\]

In light of Lemma 33 and Lemma 34, presenting a large family of subspace polynomials that agree on many top coefficients suffices for providing a word that is adjacent to too many Reed–Solomon codewords. Such a family of size \(\left[\frac{n/g}{r/g}\right] q^{r/g}\) was presented in Theorem 23, where \(g|\gcd(n, r)\) and \(\ell = \frac{n-r}{g} - 1\). Using the standard bound on the Gaussian coefficient (see Section 5.1) we have that

\[
\frac{\left[\frac{n/g}{r/g}\right] q^{r/g}}{q^{n\ell}} \leq 4q^{r(g-1)-(n-r)\ell}.
\]

Plugging in the expression for \(\ell\) results in an upper bound of \(4q^n\), and hence the size of the family is not more than 4 times the length of the code, which is \(q^n\). In addition, an explicit family can be derived from Construction 4 whose size is not super-polynomial in \(n\) either, and hence a super-polynomial list decoding bound is not achieved.
Both of these families do provide dense sets that are larger than the ones achieved by a counting argument. Dense sets of Reed–Solomon codewords have applications in hardness of approximating the minimum distance of a linear code \([8]\) and in constructing error-correcting codes with improved parameters \([32]\). However, the dense sets provided by our results are not nearly large enough for these applications.

**Appendix B**

The following lemmas shows that the bound from Theorem \([25]\) strictly outperforms the bound implied by Theorem \([24]\) and Corollary \([15]\) given in \([31]\), when applied over the code in Example \([3]\).

**Lemma 35** For any \(i \geq 1\),

\[
1 - \sqrt{\frac{2^i - 1}{2^i}} > \frac{1}{2^{i+1}}.
\]

**Proof.** Clearly, \(\frac{1}{2^i} > 0\), and hence,

\[
2^i - 1 + \frac{1}{2^i+2} > 2^i - 1
\]

\[
1 - \frac{2}{2^i+1} + \frac{1}{2^i+2} > \frac{2^i - 1}{2^i}
\]

\[
\left(1 - \frac{1}{2^i+1}\right)^2 > \frac{2^i - 1}{2^i}
\]

\[
1 - \frac{1}{2^{i+1}} > \sqrt{\frac{2^i - 1}{2^i}}
\]

\[
1 - \sqrt{\frac{2^i - 1}{2^i}} > \frac{1}{2^{i+1}}
\]

**Lemma 36** For a large enough \(n\), the radius \(\tau = \frac{n}{2^{i+1}}\), for which the code in Example \([3]\) cannot be list decoded efficiently according to Theorem \([25]\), is strictly smaller than the radius \(\tau'\) which is guaranteed by the Corollary \([15]\).

**Proof.** Inserting \(\varepsilon = 1\) into the bound of Corollary \([15]\) provides a stronger bound than Corollary \([15]\) for any \(\varepsilon < 1\). Therefore, when our
bound outperforms Corollary 15 with $\varepsilon = 1$, our bound also outperforms Corollary 15 with $\varepsilon < 1$.

Since in Example 3 we have $d = \frac{n}{2^i} - 1$ it follows that

\[
\tau' \geq \frac{m + n}{2} - \sqrt{\frac{(m+n)^2}{4} - m(d-1)}
= \frac{m + n}{2} - \sqrt{\frac{(m+n)^2}{4} - m \left( \frac{n}{2^i} - 2 \right)}
= \left( \frac{m+n}{2} \right) \left( 1 - \sqrt{1 - \frac{4m(d-1)}{(m+n)^2}} \right). 
\] (5.4)

Notice that by Theorem 24, the bound of 31 is weaker if $m > n$, whereas the bound of Theorem 25 does not depend on $m$. Therefore, it suffices to show that the bound from Theorem 25 outperforms the one from 31 for $m = n$. In this case, (5.4) simplifies to

\[
\tau' \geq n \left( 1 - \sqrt{1 - \frac{1}{2^i} + \frac{2}{n}} \right). 
\] (5.5)

For a large enough $n$ the term $\frac{2}{n}$ may be neglected. Hence, by Lemma 35 (5.5) implies that

\[
\tau' \geq n \left( 1 - \sqrt{\frac{2^i - 1}{2^i}} \right) > \frac{n}{2^{i+1}} = \tau.
\]

\[\blacksquare\]

Acknowledgment

The authors would like to thank Ron M. Roth for bringing up the idea of puncturing Gabidulin codes (Lemma 30 and Corollary 16), and the idea of Example 4.

Further, the authors would like to thank the reviewers whose comments helped to improve the presentation of the paper.
Bibliography


5. A Correction

Unfortunately, a mistake was recently found in the proof of Theorem 25. This subchapter provides a correction to it, which weakens the result. The corrected theorem applies only for a subset of the Gabidulin codes which were initially thought not to be list-decodable. In our current research we pursue a correction which applies to all Gabidulin codes which were initially stated to be not list-decodable, and furthermore, we strive to improve our results beyond what is stated in the paper.

At the end of the proof of Theorem 25, it is stated that

\[
\text{rank}(c_R - c_{R-P}) = \text{rank}((P(\alpha_1), \ldots, P(\alpha_n))) \leq \dim \text{Im}(P) = n - \dim \ker(P),
\]

in which the last equality is an incorrect statement. This is since the subspace polynomial \( P \), albeit being a polynomial over the field \( \mathbb{F}_{q^n} \), is applied
over elements of $\mathbb{F}_{q^n}$, which is an extension of $\mathbb{F}_{q^m}$. Therefore, the correct equality is $\dim \text{Im}(P) = m - \dim \ker(P)$, which is of no use for the proof of the theorem.

Clearly, the theorem holds as is for the case $n = m$, i.e., for square Gabidulin codes. Furthermore, it also holds for some cases in which $m > n$, in which the evaluation points $\alpha_1, \ldots, \alpha_n$ reside in any cyclic shift of the subfield $\mathbb{F}_{q^n}$. The corrected theorem is as follows.

**Theorem 30** For integers $k \leq n \leq m$ such that $n$ divides $m$, let $\text{Gab}[n,k]$ be a linear Gabidulin code over $\mathbb{F}_{q^m}$, with $d = n - k + 1$ and evaluation points $\alpha_1, \ldots, \alpha_n \in \beta \mathbb{F}_{q^n}$ for some $\beta \in \mathbb{F}_{q^m}^*$. Let $\tau, g$ be integers such that $\left\lfloor \frac{d-1}{2} \right\rfloor + 1 \leq \tau \leq d - 1$, $g \geq 2$, and $g \mid \gcd(n - \tau, n)$. If $\ell$ is the unique integer such that $n = n - \tau + g(\ell + 1)$ (and thus, $\tau = g(\ell + 1)$), then there exists a word $c_R \in \mathbb{F}_{q^m}^n \setminus \text{Gab}[n,k]$ such that

$$|\text{Gab}[n,k] \cap B_{\tau}(c_R)| \geq \frac{\left\lfloor \frac{n}{g} \right\rfloor}{q^\tau}.$$

**Proof.** According to Theorem 23, there exists a set $P$ of $\left\lfloor \frac{n}{g} \right\rfloor$ subspace polynomials of subspaces in $G_q(n,n - \tau)$, that agree on their topmost $\tau = g(\ell + 1)$ coefficients. The coefficients of these polynomials are in the field $\mathbb{F}_{q^n}$. Since $n|m$, we have that $\mathbb{F}_{q^n}$ is a subfield of $\mathbb{F}_{q^m}$, and thus these coefficients may be considered as elements of $\mathbb{F}_{q^m}$. Recall that according to Lemma 29, these polynomials agree on their topmost $\tau$ coefficients also when considered as polynomials over $\mathbb{F}_{q^m}$.

Further, let $\{V_P\}_{P \in P} \subseteq G_q(n,n - \tau)$ be the subspaces which correspond to the subspace polynomials in $P$. For $P \in P$, let $P_\beta$ be the subspace polynomial of $\beta V_P$, and let $P_\beta \triangleq \{P_\beta\}_{P \in P}$. According to Lemma 24 and according to the properties of $P$, it follows that the polynomials in $P_\beta$ agree on their topmost $\tau$ coefficients. Since multiplication by $\beta$ is an injection, it also follows that $|P| = |P_\beta|$.

Let $R$ be any linearized polynomial over $\mathbb{F}_{q^m}$ of $q$-degree $n - \tau$ that has the mutual top coefficients of $P_\beta$, and let $c_R \in \mathbb{F}_{q^m}^n$ be the word resulting from the evaluation of $R$ at $\alpha_1, \ldots, \alpha_n$. Similarly, for $P_\beta \in P_\beta$ let $c_{R-P_\beta} \in \mathbb{F}_{q^m}^n$ be the word corresponding to the evaluation of $R - P_\beta$ at $\alpha_1, \ldots, \alpha_n$.

Since $\deg_q(R - P_\beta) \leq n - \tau - g(\ell + 1)$ and $\tau = g(\ell + 1) > \frac{d-1}{2} = \frac{n-k}{2}$ it
follows that $2\tau = \tau + g(\ell + 1) > n - k$, and hence,

$$k > n - \tau - g(\ell + 1) \geq \deg_q(R - P_\beta).$$

Therefore, the word $c_{R - P_\beta}$ is a codeword of $\text{Gab}[n,k]$ for all $P_\beta \in \mathcal{P}_\beta$. In addition, since $\tau \leq d - 1$ it follows that $\deg_q R = n - \tau \geq n - d + 1 = k$, and hence $c_R \notin \text{Gab}[n,k]$.

Since every linearized polynomial can be viewed as an $\mathbb{F}_q$-linear mapping (see Section 5.1), it follows that for every $P_\beta \in \mathcal{P}_\beta$,

$$\rank(c_{R - P_\beta}) = \rank((P_\beta(\alpha_1), \ldots, P_\beta(\alpha_n)))$$

$$= \dim \langle P_\beta(\alpha_1), \ldots, P_\beta(\alpha_n) \rangle$$

$$= \dim P_\beta ((\alpha_1, \ldots, \alpha_n))$$

$$= \dim P_\beta(\beta \mathbb{F}_{q^n}),$$

where the last equality follows from the fact that $\alpha_1, \ldots, \alpha_n$ are $n$ linearly independent elements in $\beta \mathbb{F}_{q^n}$, a subspace of dimension $n$. Since $P_\beta$ is a subspace polynomial of $\beta V_P$, which is a subspace of dimension $n - \tau$ that is contained in $\beta \mathbb{F}_{q^n}$, it follows that $\dim P_\beta(\beta \mathbb{F}_{q^n}) = \tau$. Thus, the set \{$c_{R - P_\beta} \mid P_\beta \in \mathcal{P}_\beta \subseteq \text{Gab}[n,k]$\} is a set of size $\left\lfloor \frac{n}{g} \right\rfloor / q^{\ell}$, which is contained in a ball of radius $\tau$ around the word $c_R$. \hfill \blacksquare

As a result of Theorem 30, all the examples and statements which follow from Theorem 25 only hold for Gabidulin codes in which the evaluation points are contained in cyclic shifts of $\mathbb{F}_{q^n}$. In particular, this holds for Example 3, Example 4, Theorem 26, and Corollary 16. The results about lifted Gabidulin codes, which are given in Theorem 28 and in Theorem 29, also only hold for lifted Gabidulin codes whose evaluation points are contained in a cyclic shift of $\mathbb{F}_{q^n}$.
Part II

Conference Papers
Chapter 6

Distributed Storage Systems Based on Intersecting Subspace Codes

Netanel Raviv and Tuvi Etzion

6.A Conference Version

Abstract

Distributed storage systems based on intersecting constant dimension (equidistant) codes are presented. These intersecting codes are constructed using the Plücker embedding, which is essential in the repair and the reconstruction algorithms. These systems possess several useful properties such as high failure resilience, minimum bandwidth, low overall storage, simple algebraic repair and reconstruction algorithms, good locality, and compatibility with small fields.

6.A.1 Introduction

Let $q$ be a prime power and let $\mathbb{F}_q$ be the finite field with $q$ elements. In a distributed storage system (DSS) a file $x \in \mathbb{F}_q^B$ is stored in $n$ storage nodes, $\alpha$ information symbols in each node. The DSS is required to be resilient to
node failures, i.e., it should be possible to retrieve the data of a lost node by contacting $d$ other active nodes and downloading $\beta$ information symbols from each one of them, an operation which is called repair. In addition, a data collector (DC) should be able to rebuild the stored file $x$ by contacting $k$ active nodes, an operation which is called reconstruction. If the file is coded with an ordinary error correcting code $C$ prior to being stored in the system (usually by an MDS code \cite{8, 10, 12, 13, 14, 15, 16}), then $C$ is called the outer code, and the DSS is called the inner code.

A repair process that results in a new node which contains the exact same information as in the failed node is called an exact repair \cite{11, 15}. A repair process which is not an exact repair is called a functional repair, and it must maintain the system’s ability of repair and reconstruction (of the original file). The amount of data which is required for a repair is $d\beta$, and it is called the repair bandwidth of the code. Codes which minimize the repair bandwidth, i.e., $d\beta = \alpha$, are called Minimum Bandwidth Regenerating (MBR) codes \cite{2}. Codes which minimize $\alpha$, i.e., $\alpha = \frac{B}{k}$, are called Minimum Storage Regenerating (MSR) codes \cite{2}.

In \cite{5} a framework for a construction of a DSS code based on subspaces is given. This framework is different from the classical one. In this framework each node $i$ is associated with a subspace $U_i$ of a vector space $U = \mathbb{F}_q^B$ called the message space.

In the “storage phase”, node $i$ stores a vector $M_i \cdot x$, where $M_i$ is a
full-rank matrix whose row span \( \langle M_i \rangle \) is \( U_i \). A set of nodes is called a reconstruction set \(^1\) if their respective subspaces span the entire message space. The file \( x \) is reconstructible from a reconstruction set \( T \subseteq [n] \), where \([n] \triangleq \{1, \ldots, n\}\), by solving a linear nonsingular equation system based on \( \{M_i \cdot x\}_{i \in T} \) and \( \{M_i\}_{i \in T} \) (see Fig. 6.1). A set \( T_j \subseteq [n] \) of nodes is called a repair set for a node \( j \) if each subspace \( U_i, i \in T_j \) contains a subspace \( W_{i,j} \subseteq U_i \) such that the span of the set \( \{W_{i,j} \mid i \in T_j\} \) contains \( U_j \). The lost information \( M_j \cdot x \) may be retrieved by manipulating the rows in a linear system based on \( \{M_{i,j} \cdot x\}_{i \in T_j} \) and \( \{M_{i,j}\}_{i \in T_j} \), where \( M_{i,j} \) is a matrix whose row span is \( W_{i,j} \) (see Fig. 6.2). This framework yields algebraic repair and reconstruction algorithms. We note that in this new framework the matrices \( M_i \) have the role of the outer code in the classical framework. Hollmann [5] Example 3.2 provides an example of an equidistant subspace code for DSS in the above framework.

Besides [5], equidistant subspace codes were also observed to be useful in [7]. However, in [7] no general non-trivial construction was made, and repair/reconstruction algorithms were not discussed. Trivial equidistant subspace code was used for a DSS in [9]. This code is also known as a spread, that is, all cyclic shifts \(^2\) of some subfield of \( \mathbb{F}_{q^n} \). Although a spread provides a good locality, it is not clear how to use it in a DSS whose number

\(^1\) [5] uses the term recovery set. We use a different term for consistency.

\(^2\) A cyclic shift of a subfield \( \mathbb{F}_{q^k} \) is a set of the form \( \{\gamma v \mid v \in \mathbb{F}_{q^k}\} \) for some \( \gamma \in \mathbb{F}_{q^n} \setminus \{0\} \).
of nodes is small.

The main goal of this paper is to present a storage code with the subspace framework, which is comparable to the best codes in the classical framework. Our code has the same storage and same repair bandwidth as [5, Example 3.2], but it posses several advantages over it, such as repairing in the presence of a larger number of simultaneous failures and good locality.

A code with similar parameters may be achieved by using the Product- Matrix MBR code construction [11, Section IV]. In this construction, the file $x$ is turned into a symmetric matrix $M$, and each node stores $\psi^*_i M$, where $\psi_i$ is a row in a Vandermonde of Cauchy matrix $\Psi$. This work applies for any $d \leq k \leq n - 1$, but by setting $d = k$, we receive a code whose parameters are identical to those presented in Corollary 20 (See Section 6.A.5). Although it is not mentioned in [11], one may replace the matrix $\Psi$ in any generator matrix of a linear block code over any finite field and receive a code with the same parameters, as well as the same repair and reconstruction capabilities, as the code mentioned in Corollary 19 (Section 6.A.5). In addition, the local repair algorithm mentioned in Section 6.A.3.2 is also possible. However, the reconstruction bandwidth in our code seems to be slightly better than the one that can be achieved in [11].

We will use the equidistant subspace codes (which are also intersecting codes) from [3] as the subspaces in our DSS. Our code stores a file $x \in \mathbb{F}_q^B$, where $B = \binom{b}{2}$ for some $b \in \mathbb{N}$, in $n$ nodes, $b - 1$ field elements in each node. The user may choose any $n$ such that $b \leq n \leq \frac{q^b - 1}{q - 1}$ in correspondence with the expected number of simultaneous node failures.

Our code achive the MBR property, and may minimize the reconstruction bandwidth under the additional assumption that each node participating in a reconstruction algorithm knows the identity of the other nodes participating in this reconstruction. That is, it is possible to reconstruct $x$ by communicating $|x| = B = \binom{b}{2}$ field elements, $\frac{b}{2}$ elements from each node if $b$ is even and either $\frac{b-1}{2}$ elements or $\frac{b+1}{2}$ elements if $b$ is odd. Without this additional assumption it is possible to reconstruct $x$ by downloading a total of $b^2 - 3b + 3$ elements from $b - 1$ nodes. These code parameters hold for any field size, whereas for fields of constant size the penalty is not being able to repair (resp. reconstruct) from any set of $d$ (resp. $k$) nodes, but rather some properly chosen ones. This drawback is also apparent in some existing DSS codes [8], [9].
For the purpose of repair, the user may choose one of two possible algorithms. The first one requires that the newcomer node (newcomer, in short) will contact \( b - 1 \) active nodes and download a single field element from each one. This algorithm will minimize the repair bandwidth. The second algorithm requires downloading all data from as little as two nodes, depending on the code construction. In either of the algorithms, when operating over fields of constant size, it is not possible to contact any set of nodes, but a proper set may be easily found, and it is guaranteed to exist as long as the number of node failures does not exceed some reasonable threshold. Over large fields, any set of \( b - 1 \) nodes will suffice reconstruction, and any set of \( b \) nodes is guaranteed to have a \( b - 1 \) subset which suffices for repair.

The code has some additional useful properties. As mentioned earlier, the user may choose between a local repair (Subsection 6.A.3.2) and a minimum bandwidth repair (Subsection 6.A.3.1). In addition, it is possible to reconstruct nodes that were not previously in the system (Corollary 17), that is, once a proper set of \( b \) nodes is stored in the system by the user, the system may use repairs in order to generate additional storage nodes without any outside interference. Two additional useful properties are apparent. One is the ability to efficiently reuse the system to store a file \( y \neq x \), without having to initialize all nodes (Subsection 6.A.4.2). This property follows directly from the linear nature of our code. The second is the ability to simultaneously repair multiple node failures in parallel (Subsection 6.A.3.3).

The rest of the paper is organized as follows. A brief overview of the intersecting subspace codes from [3] will be given in Section 6.A.2. Repair and reconstruction algorithms will be discussed in Section 6.A.3 and Section 6.A.4 respectively. In Section 6.A.5 we discuss our choice for the number of vertices in the system. Conclusions will be given in Section 6.A.6. For lack of space, proofs and further explanations in this version are omitted and will appear in the full version of this paper [4], where currently, an earlier and slightly weaker construction is available.

### 6.A.2 Preliminaries

The Grassmannian \( \mathcal{G}_q(m, \ell) \) is the set of all \( \ell \)-subspaces of \( \mathbb{F}_q^m \). The size of \( \mathcal{G}_q(m, \ell) \) is given by the Gaussian coefficient \( \binom{m}{\ell}_q \) (see [6], Chapter 24). A constant dimension code (CDC) is a subset of \( \mathcal{G}_q(m, \ell) \) with respect to
the subspace metric \(d_S(U, V) = \dim U + \dim V - 2 \dim(U \cap V)\). A CDC is called equidistant if the distance between any two distinct codewords is some fixed constant. An equidistant CDC is also called a \(t\)-intersecting code since the dimension of the intersection of any two distinct codewords is some constant \(t\). Our construction uses the 1-intersecting equidistant subspace codes from [3], whose construction and properties are hereby described. Note that the intersecting property of these codes is extensively used throughout our discussion, whereas the distance property is not used at all.

In what follows \(e_i\) denotes the \(i\)th unit vector. For a set \(S\) of vectors, \(\langle S \rangle\) denotes the linear span of \(S\), and for a matrix \(M\), \(\langle M \rangle\) denotes its row linear span.

**Definition 12** (The Plücker embedding, see [1, p. 165]) Given \(M \in \mathbb{F}^{t \times b}\), identify the coordinates of \(\mathbb{F}^{(b)}_q\) with all \(t\)-subsets of \([b]\), and define \(\varphi(M)\) as a vector of length \((b)\) in which

\[
(\varphi(M))_{\{i_1, \ldots, i_t\}} \triangleq \det (i_1, \ldots, i_t)
\]

where \(M (i_1, \ldots, i_t)\) is the \(t \times t\) sub-matrix of \(M\) formed from columns \(i_1 < \ldots < i_t\). For \(v, u \in \mathbb{F}^b\) we denote by \(\varphi(v, u)\) the result of applying \(\varphi\) on the \(2 \times b\) matrix \((v^t_u)\).

**Definition 13** [3, Subsection 3.1] For \(V \in \mathcal{G}_q(b, 1)\), \(v \in V \setminus \{0\}\), and the index \(r(v)\) of the leftmost nonzero entry of \(v\), let

\[
P_V \triangleq \left( \{ \varphi(v, e_i) \}_{i \in [b] \setminus \{r(v)\}} \right).
\]

By the properties of the determinant function, any choice of a nonzero vector \(v\) from the 1-subspace \(V\) results in the same subspace, and thus \(P_V\) is well-defined. This definition can be easily generalized by using the results of Lemma [38] which follows.

**Theorem 31** [3, Theorem 14] The code

\[
C \triangleq \{ P_V \mid V \in \mathcal{G}_q(b, 1) \},
\]

is a subset of \(\mathcal{G}_q\left(\binom{b}{2}, b - 1\right)\). \(C\) is a 1-intersecting code of size \(\binom{b}{1}_q\), that is, any distinct \(P_U, P_V \in C\) satisfy \(\dim(P_U \cap P_V) = 1\). In addition, for every
distinct \( P_U, P_V \in \mathbb{C} \), \( P_U \cap P_V = \langle \varphi(u,v) \rangle \), where \( U = \langle u \rangle \) and \( V = \langle v \rangle \).

The following lemma shows that the function \( \varphi \) from Definition \ref{def:distphi} is a bilinear form when applied on two row matrices. This fact will be prominent in our constructions.

**Lemma 37** [Lemma 4] If \( v, u \in \mathbb{F}_b^q \) are nonzero vectors, and \( \gamma, \delta \in \mathbb{F}_q \), then
\[
\varphi(v, \gamma u + \delta w) = \gamma \cdot \varphi(v, u) + \delta \cdot \varphi(v, w) \quad \text{and} \quad \varphi(\gamma u + \delta w, v) = \gamma \cdot \varphi(u, v) + \delta \cdot \varphi(w, v).
\]

Lemma 38 and Lemma 39, which follow, provide a convenient way of choosing a basis to any \( P_V \in \mathbb{C} \) (Theorem \ref{thm:distphi}) and to the entire space \( \mathbb{F}_B^q \). Both lemmas will play an important role in the repair and reconstruction algorithms.

**Lemma 38** If \( v \in \mathbb{F}_b^q \) is nonzero, \( \{u_i\}_{i \in [b]} \) is a basis for \( \mathbb{F}_b^q \) and 
\[
v = \sum_{i \in [b]} \gamma_i u_i \quad \text{then each} \quad j \in [b] \quad \text{such that} \quad \gamma_j \neq 0 \quad \text{satisfies} \quad P_{(v)} = \langle \varphi(v, u_i) \rangle_{i \in [b] \setminus \{j\}}.
\]

**Lemma 39** If \( \{u_i\}_{i \in [b]} \) is a basis for \( \mathbb{F}_b^q \) then \( \{\varphi(u_i, u_j)\}_{i \neq j} \) is a basis for \( \mathbb{F}_B^q \).

We are now in a position to describe the construction of our DSS. Each storage node will be identified by a vector \( v \in \mathbb{F}_b^q \). The feasibility of the described repair and reconstruction algorithms will depend on a certain assignment of vectors. Different assignments and their resulting parameters will be discussed separately in Section \ref{sec:distphi}. With respect to a certain assignment of vectors to nodes, we will say that a set of nodes are linearly independent if their assigned vectors are linearly independent.

Let \( v_1, \ldots, v_n \) be the available storage nodes. We identify each \( v_i \) by a normalized vector from \( \mathbb{F}_b^q \); that is, a vector whose leftmost nonzero entry \( r(v_i) \) is 1. Let \( M_{v_i} \) be the \((b-1) \times B\) matrix whose rows are the vectors
\[
\{\varphi(v_i, e_j)\}_{e_j \in [b] \setminus r(v_i)}.
\]
Following the terminology in [5, Section III-A.], each node \( v_i \) is in fact associated with a subspace. In our system, this subspace is \( P_{(v_i)} = \langle M_{v_i} \rangle \) (see Definition \ref{def:subspace}).

Let \( s \) be the source node, i.e. the node holding the file \( x \in \mathbb{F}_b^q \) to be stored. For the initial storage, \( s \) sends \( M_{v_i} \cdot x \) to node \( v_i \) for all \( i = 1, \ldots, n \).
Notice that nodes whose corresponding identifier vectors are unit vectors are systematic nodes up to multiplication by \( \pm 1 \), that is, each element they hold is either an entry of \( x \) or a negation of an entry of \( x \) (see Fig. 6.3). Notice that the vectors \( \varphi(v_i, e_j) \) are sparse, that is, they have at most \( b \) nonzero entries. This fact is useful for reducing the space and time complexity of the storage phase.

<table>
<thead>
<tr>
<th>Storage Node no. ( i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identifier: ( v_i \in \mathbb{F}_q^b )</td>
</tr>
<tr>
<td>Associated Subspace: ( P_{(v_i)} )</td>
</tr>
<tr>
<td>Stored Data</td>
</tr>
<tr>
<td>( \varphi(v_i, e_1) \cdot x )</td>
</tr>
<tr>
<td>Without ( \varphi(v_i, e_j) \cdot x ) for ( j = r(v_1) ).</td>
</tr>
</tbody>
</table>

Figure 6.3: The structure of a storage node.

6.A.3 Repair

6.A.3.1 Minimum Bandwidth Repair

In what follows we show that it is possible to repair a node failure by communicating a single field element from \( b - 1 \) nodes. For functional repair no further computations are needed while for exact repair an additional \( O(B^2) \) algorithm should be applied by the newcomer.

**Algorithm 1** Let \( v_j \) be the failed node, let \( \{u_i\}_{i \in [b]} \) be a basis of \( \mathbb{F}_q^b \) of active nodes, and \( v = \sum_{i \in [b]} \gamma_j u_j \). The newcomer picks any \( t \in [b] \) such that \( \gamma_t \neq 0 \), and downloads the element \( \varphi(v_j, u_t) \cdot x \) from each of the nodes \( \{u_i\}_{i \in [b] \setminus \{t\}} \).

Notice that for each \( i \in [b] \), \( \varphi(v_j, u_i) \in P_{(u_i)} \) and thus Algorithm 1 is well defined.
Corollary 17 Using Algorithm 1, it is possible to add a new node that was not initially in the DSS.

The following Lemma follows from Lemma 38.

Lemma 40 By using the information received from Algorithm 1, the newcomer may restore the information from the failed node $v_j$ by using $O(B^2)$ field operations. For functional repair, no further computations are needed.

6.A.3.2 Local Repair

It is often required that a failed node will be repairable from as few other active nodes as possible. It is clear that without replication of nodes, a minimum of two active nodes is necessary for such a repair. In the sequel we present an alternative repairing approach that may achieve this minimum. The possibility of achieving this minimum depends on the specific assignment of vectors to the nodes, as will be discussed in Section 6.A.5.

Algorithm 2 Let $v_j$ be the failed node and let $\{u_1, \ldots, u_\ell\}$ be a set of active linearly independent nodes such that $v_j \in \langle u_1, \ldots, u_\ell \rangle$. For all $t \in [\ell]$, the newcomer downloads the entire vector $M_{u_t} \cdot x$ from $u_t$.

Lemma 41 By using the information received from Algorithm 2, the newcomer may restore the information of the failed node $v_j$ in $O(\ell^2 \cdot b)$ field operations.

Corollary 18 Let $v_j$ be a failed node. If $\ell$ is the smallest integer such $v_j$ is in the linear span of $\ell$ other active nodes, then the locality of repairing $v_j$ is $\ell$.

The minimum values of $\ell$ are discussed in Section 6.A.5.

6.A.3.3 Parallel Repair

Consider the scenario of multiple simultaneous node failures. Obviously, under $t$ failures, if the conditions of Algorithm 1 or Algorithm 2 are satisfied, then it is possible to execute $t$ sequential instances of the repair algorithm. We show that the local repair (Algorithm 2) could be improved in this sense in a certain special cases. This is a simple consequence of Lemma 41.
Lemma 42 If \( \{v_{i_1}, \ldots, v_{i_t}\} \) is a set of failed nodes and \( \{v_{j_1}, \ldots, v_{j_s}\} \) is a set of active linearly independent nodes, of the remaining nodes, such that

\[
\{v_{i_1}, \ldots, v_{i_t}\} \subseteq \langle v_{j_1}, \ldots, v_{j_s}\rangle,
\]

then it is possible to repair all failures by communicating \( s \cdot (b - 1) \) field elements.

The complexity of repairing in parallel remains \( t \) times the complexity of Algorithm 2. However, the amount of communication is the same as in a single instance of Algorithm 2. It is evident that this algorithm requires good locality. An assignment of vectors to nodes that achieves good locality is discussed in Section 6.A.5.

6.A.4 Reconstruction and Modification

6.A.4.1 Reconstruction

In this subsection presents two reconstruction algorithms for two different models of communication. In Algorithm 3, which follows, the DC accesses \( b - 1 \) active nodes and downloads a total of \( b^2 - 3b + 3 \) field elements.

Using the additional assumption that the nodes participating in the reconstruction know the identities of one another (e.g., by broadcast, shared memory, or by acknowledgement from the DC), we show that it is possible to reconstruct \( \mathbf{x} \) by communicating \( B \) field elements. This is the minimum communication that guarantees a complete reconstruction of \( \mathbf{x} \).

Algorithm 3 Let \( \{u_1, \ldots, u_{b-1}\} \) be any set of active linearly independent nodes. Let \( u_b \) be a vector, not necessarily affiliated with an active node, which completes \( \{u_1, \ldots, u_{b-1}\} \) to a basis of \( F_q^b \). For each \( j \in [b-2] \), the DC downloads the vector \( M_{u_j} \cdot \mathbf{x} \) from \( u_j \). In addition, the DC downloads the element \( \varphi(u_{b-1}, u_b) \) from \( u_{b-1} \). The DC assembles the vector \( \mathbf{w} \in F_q^B \) such that \( \mathbf{w}_{i,j} = \varphi(u_i, u_j) \cdot \mathbf{x} \), and the \( B \times B \) matrix \( A \) whose rows are \( \{\varphi(u_i, u_j)\}_{i \neq j} \). The vector \( \mathbf{x} \) is then reconstructed by solving the linear system of equations \( A \mathbf{x} = \mathbf{w} \).

\(^3\)The entries of the vector \( \mathbf{w} \in F_q^B \) are identified by all 2-subsets of \( [b] \) according to the lexicographic order.
Lemma 43 The matrix $A$ in Algorithm 3 has full rank. In particular, the DC may extract $x$ by performing $O(B^3)$ field operations and downloading $(b - 2)(b - 1) + 1 = b^2 - 3b + 3$ field elements.

By Lemma 39 it is evident that necessary and sufficient information for the reconstruction of $x$ is $\{\varphi(u_i, u_j) \cdot x\}_{i \neq j}$ for any basis $\{u_1, \ldots, u_b\}$. Therefore, assuming that each node participating in a reconstruction algorithm knows the identity of the other nodes participating in this reconstruction, it is possible to reduce the communication to merely $|x| = B$ field elements from $b - 1$ nodes. This will be done by using a binary $b \times b$ matrix, denoted $N$, which is an anti-symmetric matrix whose sum of entries is $B$. The $i, j$ entry of $N$ indicates if the element $\varphi(u_i, u_j) \cdot x$ should be downloaded from node $u_i$ or from node $u_j$. An explicit construction of $N$ for all values of $b$ appears in [4].

Lemma 44 Assuming the nodes participating in the reconstruction know the identity of each other, the DC may reconstruct $x$ by performing $O(B^3)$ field operations and downloading $B$ field elements. This will be done by downloading $\varphi(u_i, u_j) \cdot x$ from $u_i$ if and only if $N_{i,j} = 1$.

6.A.4.2 Modification

A useful property of a DSS is being able to update a small fraction of $x$ without having to initialize the entire system. The linear nature of our code and the absence of an outer code allows these modifications to be done efficiently. In particular, the complexity of the process is a function of the Hamming distance $d_H(x, y)$, where $y$ is the modification of the vector $x$. In an MDS based distributed storage systems a change of a single bit of $x$ usually requires changing a large portion of the data. Therefore, one more advantage of our system is revealed.

Lemma 45 If $x \in \mathbb{F}_q^B$ is stored in the system, it is possible to update the system to contain $y \in \mathbb{F}_q^B$ by communicating $(\log B + \log q) \cdot d_H(x, y) \cdot n$ bits.

6.A.5 Assignment of Vectors

In Section 6.A.3 and Section 6.A.4 we proved that the performance of the detailed algorithms strongly relies on the chosen vectors $v_1, \ldots, v_n$. Since both
repair and reconstruction algorithms require linearly independent nodes, it follows that the assigned set of vectors should contain a basis of $\mathbb{F}_q^b$ even after multiple failures.

**Definition 14** For $t \in \mathbb{N}$ a set $S \subseteq \mathbb{F}_q^b$ is called a $t$-resilient spanning set if each $t$-subset $T$ of $S$ satisfies $\langle S \setminus T \rangle = \mathbb{F}_q^b$.

**Observation 1** If $S$ is a $t$-resilient spanning set then by using $|S|$ storage nodes assigned with the vectors in $S$, it is possible to repair and reconstruct in the presence of up to $t$ simultaneous node failures.

In what follows we present a construction of a set of vectors $\{v_1, \ldots, v_n\}$ compatible with Algorithm 1 achieving the MBR property $d\beta = \alpha$. This construction will rely on a well-known property of a generator matrix of a linear code.

**Lemma 46** If $C$ is a linear block code over $\mathbb{F}_q$ of length $n$, dimension $b$, distance $\delta$, and generator matrix $G$ (that is, $C \triangleq \{xG | x \in \mathbb{F}_q^b\}$), then the columns of $G$ are a $(\delta - 1)$-resilient spanning set.

**Corollary 19** If $C$ is a linear block code over $\mathbb{F}_q$ of length $n$, dimension $b$, distance $\delta$, and generator matrix $G$, then by assigning the columns of $G$ to nodes in the DSS, under any $\delta - 1$ node failures, there exists a set of active linearly independent nodes from which repair and reconstruction are possible.

In any linear code, the so called Singleton bound states that $b \leq n - \delta + 1$, and hence a set $A$ of $n - \delta + 1$ nodes is not necessarily a basis to $\mathbb{F}_q^b$. Therefore, the actual set of nodes required for repair and reconstruction is some $b$-subset of $A$. Notice that although this $b$-subset is guaranteed to exist by Lemma 46, not every $b$-subset of $A$ is necessarily a basis. However, when the code $C$ is an MDS code (that is, $b = n - \delta + 1$) we get the following simple corollary.

**Corollary 20** If $C$ is an MDS code over $\mathbb{F}_q$ with length $n$, dimension $b$, distance $\delta$, and generator matrix $G$, then by assigning the columns of $G$ to nodes in the DSS, any set of $b$ active nodes allows repair and reconstruction.
Notice that Corollary 19 holds for any \( q \), whereas Corollary 20 requires the existence of an MDS code, and hence a large \( q \) is required [17].

Algorithm 2 in Subsection 6.A.3.2 may possibly achieve the optimal locality. It is evident from Lemma 41 that in order to have good locality, the set \( \{u_1, \ldots, u_\ell\} \) from Algorithm 2 is required to be small. We show that by choosing some basis of \( \mathbb{F}_q^b \), partitioning it to equally sized subsets and taking the linear span of each subset, some locality is achievable. The resulting failure resilience will grow with the field size. Thus, this technique will be particularly useful in large fields.

**Definition 15** Let \( c \) be a positive integer such that \( c \) divides \( b \), and let \( A = \{v_1, \ldots, v_b\} \) be a basis of \( \mathbb{F}_q^b \). Partition \( A \) into \( \frac{b}{c} \) equally sized subsets \( A_i = \{v_i, v_{i+1}, \ldots, v_{(i+1)c}\} \) for \( i \in \{0, \ldots, \frac{b}{c} - 1\} \). Let \( V_i \subseteq \mathbb{F}_q^b \) be a set of \( \left[ \frac{c}{q} \right] \) representatives for the 1-subspaces of \( \langle A_i \rangle \). Finally, let \( V = \bigcup_{i=1}^{b/c} V_i \).

**Lemma 47** The set \( V \) from Definition 15 is a \((q^{c-1} - 1)\)-resilient spanning set (see Definition 14). Furthermore, assigning \( V \) to nodes in a DSS allows repairing any node failure using at most \( c \) active nodes.

### 6.A.6 Conclusions

A construction of a DSS code based on intersecting constant dimension codes was presented. These were shown to have several desired properties such as high failure resilience, minimum bandwidth, low overall storage, simple algebraic repair and reconstruction algorithms, good locality, and compatibility with small fields. For future research, we would like to have subspace based MBR codes for all feasible parameters, as well as MSR codes. We would also like to have an assignment of vectors which enables both local repair and minimum bandwidth repair, for instance, by using a generator matrix of a locally decodable code.

### Acknowledgments

This work was supported in part by the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant no. 10/12.
Bibliography


Abstract

Distributed storage systems based on equidistant constant dimension codes are presented. These equidistant codes are based on the Plücker embedding, which is essential in the repair and the reconstruction algorithms. These systems posses several useful properties such as high failure resilience, minimum bandwidth, low overall storage, simple algebraic repair and reconstruction algorithms, good locality, and compatibility with small fields.

6.B.1 Introduction

Let $q$ be a prime power and let $\mathbb{F}_q$ be the field with $q$ elements. In a distributed storage system (DSS) a file $x \in \mathbb{F}_q^B$ is stored in $n$ storage nodes, $\alpha$ information symbols in each. The DSS is required to be resilient to node failures; i.e., it should be possible to retrieve the data from a lost node by contacting $d$ other active nodes and downloading $\beta$ information symbols from each one of them, an operation which is called repair. In addition, a data collector (DC) should be able to rebuild the stored file $x$ by contacting $k$ active nodes, an operation which is called reconstruction. If the file is coded with an ordinary error correcting code $C$ prior to being stored in the system (usually by an MDS code [2, 3, 4, 5, 6, 7, 8]), then $C$ is called the outer code, and the DSS is called the inner code.

A repair process that results in a new node which contains the exact same information as in the failed node is called an exact repair [2, 9]. A repair process which is not an exact repair is called a functional repair, which must maintain the system’s ability of repair and reconstruction (of the original file). The amount of data which is required for a repair is $d\beta$, and it is called the repair bandwidth of the code. Codes which minimize the repair bandwidth, i.e., $d\beta = \alpha$, are called Minimum Bandwidth Regenerating (MBR) codes [10]. Codes which minimize $\alpha$, i.e., $\alpha = \frac{B}{k}$, are called Minimum Storage Regenerating (MSR) codes [10]. A Self-Repairing Code (SRC) [11] is a code satisfying: (a) repairs are possible without having to download an amount of data equivalent to the reconstruction of the original file $x$; and (b) the number of nodes required for repair depends only on how
many nodes are missing and not on their identity.

In [12] a framework for a construction of a DSS code based on subspaces is given. This framework is slightly different from the classical one. In this framework every node $v_i$ is associated with a subspace $U_i$ of a vector space $U = \mathbb{F}_q^B$ called the message space. In the “storage phase” a node $v_i$ stores a vector $M_i \cdot x$, where $M_i$ is a full-rank matrix whose row span is $U_i$. A set of nodes is called a reconstruction set if their respective subspaces span the entire message space. The file $x$ is reconstructible from a reconstruction set $\{v_i\}_{i \in T}$, $T \subseteq [n]$, by solving a linear nonsingular equation system based on $\{M_i \cdot x\}_{i \in T}$ and $\{M_i\}_{i \in T}$. A set $\{v_i\}_{i \in T_j}, T_j \subseteq [n]$ of nodes is called a repair set for a node $v_j$ if each subspace $U_i, i \in T_j$ contains a subspace $W_{i,j} \subseteq U_i$ such that the span of the set $\{W_{i,j} | i \in T_j\}$ contains $U_j$. The lost information $M_j \cdot x$ may be retrieved by manipulating the rows in a linear system based on $\{M_{i,j} \cdot x\}_{i \in J}$ and $\{M_{i,j}\}$, where $M_{i,j}$ is a matrix whose row span is $W_{i,j}$. This framework yields an algebraic repair and reconstruction algorithms. We note that in this new framework the matrices $M_i$ have the role of the outer code in the classical framework.

We will use the equidistant subspace codes from [13] as the subspaces in our DSS. Our code stores a file $x \in \mathbb{F}_q^B$, where $B = \binom{b}{2}$ for some $b \in \mathbb{N}$, in $n$ nodes, $b - 1$ field elements in each node. The user may choose any $n$ such that $b \leq n \leq \frac{b^2-1}{q-1}$ in correspondence with the expected number of simultaneous node failures.

Our codes achieve the SRC and MBR properties, and minimize the reconstruction bandwidth under the additional assumption that the nodes participating in the reconstruction algorithm know the identity of one another. That is, it is possible to reconstruct $x$ by communicating $|x| = B = \binom{b}{2}$ field elements, $\frac{b}{2}$ elements from each node if $b$ is even and either $\frac{b-1}{2}$ elements or $\frac{b+1}{2}$ elements if $b$ is odd. Without this additional assumption it is possible to reconstruct $x$ by downloading $b^2 - 3b + 3$ elements from $b - 1$ nodes. These code parameters hold for any field size, whereas for fields of constant size the penalty is not being able to repair (resp. reconstruct) from any set of $d$ (resp. $k$) nodes, but rather some properly chosen ones. This drawback is also apparent in some existing DSS codes [3, 11].

For the purpose of repair, the user may choose one of two possible algorithms. The first one requires that the newcomer node (newcomer, in short)

\[12\] uses the term recovery set. We use a different term for consistency.
will contact either \( b - 1 \) active nodes and download a single field element from each one. This algorithm will minimize the repair bandwidth as possible. The second algorithm requires downloading all data from as little as two nodes, depending on the code construction. When operating over fields of constant size, it is not possible to repair from any set of \( b - 1 \) nodes, but a proper set may be easily found, and it is guaranteed to exist as long as the number of node failures does not exceed some reasonable threshold. Over large fields, any set of \( b - 1 \) nodes will suffice for repair and reconstruction.

The presented code has several useful properties. As mentioned earlier the user may choose between a local repair (Subsection 6.B.3.3) and a minimum bandwidth repair (Subsection 6.B.3.2). In addition, it is possible to reconstruct nodes that were not previously in the system (Corollary 21); that is, once a proper set of \( b \) nodes is stored in the system by the user, the system may use repairs in order to generate additional storage nodes without any outside interference. It is also possible to repair in the presence of up to \( O(\sqrt{B}) \) simultaneous node failures, while imposing no restriction on the field size (Example 7). Two additional useful properties are apparent. One is the ability to efficiently reuse the system to store a file \( y \neq x \), without having to initialize all nodes (Subsection 6.B.3.6). This property follows directly from the linear nature of our code. The second is the ability to simultaneously repair multiple node failures in parallel (Subsection 6.B.3.4).

A brief overview of the equidistant subspace codes from [13] will be given in Section 6.B.2. The specific properties of our code strongly depend on an assignment of different vectors as identifiers to the storage nodes. The code will first be described with respect to a general assignment in Section 6.B.3, and specific assignments, as well as their resulting properties, will be discussed in Section 6.B.4. A code with identical parameters, similar repair and reconstruction capabilities, larger reconstruction bandwidth, and a fundamentally different construction was presented in [9]. A more detailed comparison with this code and other codes is found in Section 6.B.5.

### 6.B.2 Preliminaries

The Grassmannian \( \mathcal{G}_q (m, \ell) \) is the set of all \( \ell \)-subspaces of \( \mathbb{F}_q^m \). The size of \( \mathcal{G}_q (m, \ell) \) is given by the Gaussian coefficient \( \binom{m}{\ell}_q \) (see [14], Chapter 24). A constant dimension code (CDC) is a subset of \( \mathcal{G}_q (m, \ell) \) with respect to
the subspace metric \(d_S(U, V) = \dim U + \dim V - 2 \dim(U \cap V)\). A CDC is called equidistant if the distance between every two distinct codewords is some fixed constant. An equidistant CDC is also called a \(t\)-intersecting code since the dimension of the intersection of any two distinct codewords is some constant \(t\). Our construction uses the 1-intersecting equidistant subspace codes from [13], whose construction and properties are hereby described.

In what follows \(e_i\) denotes the \(i\)th unit vector. For a set \(S\) of vectors, \(\langle S \rangle\) denotes the linear span of \(S\), and for a matrix \(M\), \(\langle M \rangle\) denotes its row linear span.

**Definition 16** (The Plücker embedding, see [3, Section 4], [1, p. 165])

Given \(M \in \mathbb{F}_q^{t \times b}\), identify the coordinates of \(\mathbb{F}_q^t\) with all \(t\)-subsets of \([b]\), and define \(\varphi(M)\) as a vector of length \(\binom{b}{t}\) in which

\[
(\varphi(M))_{\{i_1, \ldots, i_t\}} \triangleq \det M(i_1, \ldots, i_t)
\]

where \(M(i_1, \ldots, i_t)\) is the \(t \times t\) sub-matrix of \(M\) formed from columns \(i_1 < \ldots < i_t\). For \(v, u \in \mathbb{F}_q^b\) we denote by \(\varphi(v, u)\) the result of applying \(\varphi\) on the \(2 \times b\) matrix \(\begin{pmatrix} v \\ u \end{pmatrix}\).

**Definition 17** [3, Subsection 3.1] For \(V \in \mathcal{G}_q(b, 1)\), \(v \in V \setminus \{0\}\), and the index \(r(v)\) of the leftmost nonzero entry of \(v\), let

\[
P_V \triangleq \left\langle \{\varphi(v, e_i)\}_{i \in [b] \setminus \{r(v)\}} \right\rangle.
\]

By the properties of the determinant function, any choice of a nonzero vector \(v\) from the 1-subspace \(V\) results in the same subspace \(P_V\), and thus \(P_V\) is well-defined. Lemma 50 which follows generalizes this definition.

**Theorem 32** [3, Theorem 14] The following code

\[
\mathcal{C} \triangleq \{P_V \mid V \in \mathcal{G}_q(b, 1)\},
\]

\(\mathcal{C} \subseteq \mathcal{G}_q\left(\binom{b}{2}, b - 1\right)\) is an equidistant 1-intersecting code of size \(\binom{b}{1}\); that is, any distinct \(P_U, P_V \in \mathcal{C}\) satisfy \(\dim(P_U \cap P_V) = 1\). In addition, for every distinct \(P_U, P_V \in \mathcal{C}\), \(P_U \cap P_V = \langle \varphi(u, v) \rangle\), where \(U = \langle u \rangle\) and \(V = \langle v \rangle\).
bilinear form when applied on two row matrices. This fact will be prominent in our constructions.

**Lemma 48**  
If $v, u \in \mathbb{F}_q^b$ are nonzero vectors, and $\gamma, \delta \in \mathbb{F}_q$, then $\varphi(v, \gamma u + \delta w) = \gamma \cdot \varphi(v, u) + \delta \cdot \varphi(v, w)$ and $\varphi(\gamma u + \delta w, v) = \gamma \cdot \varphi(u, v) + \delta \cdot \varphi(w, v)$.

Lemmas 49, 50 and 51 which follow, provide a convenient way of choosing a basis to any $P_V \in \mathbb{C}$ (Theorem 32) and to the entire space $\mathbb{F}_q^B$. These lemmas will play an important role in the repair and reconstruction algorithms.

**Lemma 49**  
If $v \in \mathbb{F}_q^b$ is nonzero and $u \in \mathbb{F}_q^b$ then $\varphi(v, u) \in P_{\langle v \rangle}$.

**Proof.** According to Lemma 48, if $v = (\gamma_1, \ldots, \gamma_b)$, then

$$\sum_{i=1}^{b} \gamma_i \varphi(v, e_i) = \varphi \left( v, \sum_{i=1}^{b} \gamma_i e_i \right) = \varphi(v, v) = 0.$$  

(6.1)

By Definition 47, $\varphi(v, e_i) \in P_{\langle v \rangle}$ for all $i \in [b] \setminus r(v)$, and thus $\varphi(v, e_i) \in P_{\langle v \rangle}$ for all $i \in [b]$ by (6.1). Hence, if $u = (\delta_1, \ldots, \delta_b)$ then

$$\varphi(v, u) = \varphi \left( v, \sum_{i=1}^{b} \delta_i e_i \right) = \sum_{i=1}^{b} \delta_i \varphi(v, e_i) \in P_{\langle v \rangle}.$$  

**Lemma 50**  
If $v \in \mathbb{F}_q^b$ is nonzero, $\{u_i\}_{i \in [b]}$ is a basis for $\mathbb{F}_q^b$ and $v = \sum_{i \in [b]} \gamma_i u_i$ then all $j \in [b]$ such that $\gamma_j \neq 0$ satisfy $P_{\langle v \rangle} = \langle \varphi(v, u_i) \rangle_{i \in [b] \setminus \{j\}}$.

**Proof.** According to Lemma 49 we have that $\varphi(v, u_i) \in P_{\langle v \rangle}$ for all $i \in [b]$. Therefore, it is sufficient to prove that for all $j \in [b]$ such that $\gamma_j \neq 0$, the set $\{\varphi(v, u_i)\}_{i \neq j}$ is linearly independent, since this set contains $b - 1$ vectors. Let $j \in [b]$ such that $\gamma_j \neq 0$, and assume for contradiction that $\sum_{i \in [b] \setminus \{j\}} \beta_i \varphi(v, u_i) = 0$ for some $\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots, \beta_b \in \mathbb{F}_q$ which are not all zeros. By Lemma 48 it follows that $\varphi(v, \sum_{i \neq j} \beta_i u_i) = 0$. Therefore, we have either that $\sum_{i \neq j} \beta_i u_i = 0$ or that $v$ and $\sum_{i \neq j} \beta_i u_i$ are nonzero and are linearly dependent. Since $\{u_i\}_{i \neq j}$ is a subset of a basis, it is an independent set, and since $\beta_1, \ldots, \beta_{j-1}, \beta_{j+1}, \ldots, \beta_b$ are not all zeros, only
the latter option is possible. Therefore, there exists a nonzero \( \delta \in \mathbb{F}_q \) such that

\[
\delta v = \sum_{i \neq j} \beta_i u_i \\
\sum_{i=1}^{b} \delta \gamma_i u_i = \sum_{i \neq j} \beta_i u_i \\
\delta \gamma_j u_j + \sum_{i \neq j} (\delta \gamma_i - \beta_i) u_i = 0.
\]

Since \( \{u_i\}_{i \in [b]} \) is a basis for \( \mathbb{F}_q^{b} \), it follows that all coefficients in this linear combination are zero, and in particular \( \delta \gamma_j = 0 \). However, \( \delta \neq 0 \) and \( \gamma_j \neq 0 \), a contradiction. \[\square\]

**Lemma 51** If \( \{u_i\}_{i \in [b]} \) is a basis for \( \mathbb{F}_q^{b} \) then \( \{\varphi(u_i, u_j)\}_{i \neq j} \) is a basis for \( \mathbb{F}_q^{B} \).

**Proof.** By Lemma 49 and Lemma 50 and since \( \{u_i\}_{i \in [b]} \) is basis, it follows that for all \( i \in [b] \),

\[
\langle \{\varphi(u_i, u_j)\}_{j \in [b]} \rangle = \langle \{\varphi(u_i, e_j)\}_{j \in [b]} \rangle.
\]

Similarly, for all \( i \in [b] \)

\[
\langle \{\varphi(e_j, e_i)\}_{j \in [b]} \rangle = \langle \{\varphi(e_j, e_i)\}_{j \in [b]} \rangle,
\]

and therefore for all \( r, \ell \in [b] \) we have that \( \varphi(e_r, e_\ell) \in \langle \{\varphi(u_i, u_j)\}_{i \neq j} \rangle \). Since clearly, \( \mathbb{F}_q^{B} = \langle \{\varphi(e_i, e_j)\}_{i \neq j} \rangle \), the claim follows. \[\square\]

The following observation will be repeatedly used throughout our algorithms.

**Observation 2** Let \( s, \ell \in \mathbb{N} \) and let \( M_1, M_2 \in \mathbb{F}_q^{s \times \ell} \) be two distinct row-equivalent matrices. If \( r_1, \ldots, r_t \) is the series of row operations that transform \( M_1 \) to \( M_2 \), then for any \( x \in \mathbb{F}_q^\ell \) it is possible to compute \( M_2x \) given \( M_1x \) and \( r_1, \ldots, r_t \).

**Proof.** Let \( E_1, \ldots, E_t \) be the invertible matrices corresponding to the row operations that transform \( M_1 \) to \( M_2 \); that is, \( E_1 \cdot E_2 \cdot \ldots \cdot E_t \cdot M_1 = M_2 \).
The claim follows directly from the fact that for any $x \in \mathbb{F}_q^\ell$, $E_1 \cdot E_2 \cdot \ldots \cdot E_t \cdot M_1 x = M_2 x$. ■

**Remark 8** The complexity analysis of Algorithms 4 through 7 in the sequel, relies mostly on the complexity of solving a system of linear equations over a finite field. This can be done either by a school book Gaussian elimination or by employing one of many faster algorithms (see [16] and references therein). However, to simplify the discussion we analyze our algorithms by using simple Gaussian elimination.

### 6.B.3 The Distributed Storage System

We are now in a position to describe the construction of the DSS. The feasibility of the described repair and reconstruction algorithms will depend on a certain assignment of vectors in $\mathbb{F}_q^b$ to identify the storage nodes. Different assignments and their resulting parameters will be discussed separately in Section 6.A.5. With respect to a certain assignment of vectors to nodes, we will say that a set of nodes are *linearly independent* if their assigned vectors are linearly independent.

#### 6.B.3.1 Storage

Let $v_1, \ldots, v_n$ be the available storage nodes. We identify each $v_i$ by a *normalized* vector from $\mathbb{F}_q^b$: that is, a vector whose leftmost nonzero entry $r(v_i)$ is 1. Let $M_{v_i}$ be the $(b - 1) \times B$ matrix whose rows are the vectors

$$\{\varphi(v_i,e_j)\}_{e_j \in [B] \setminus r(v_i)}.$$

Following the terminology in [12], Section III.A., each node $v_i$ is in fact associated with a subspace. In our system, this subspace is $P_{(v_i)} = \langle M_{v_i} \rangle$ (see Definition 17). Let $s$ be the source node, i.e. the node holding the file $x \in \mathbb{F}_q^B$ to be stored. For the initial storage, $s$ sends $M_{v_i} \cdot x$ to $v_i$ for all $i = 1, \ldots, n$. Notice that nodes whose corresponding vector is a unit vector are systematic nodes up to multiplication by $\pm 1$, that is, each element they hold is either an entry of $x$ or a negation of an entry of $x$.

It is evident that $n \cdot (b - 1)$ field elements are being sent and stored. As for time complexity, computing the product $M_{v_i} \cdot x$ requires computing
the matrix $M_{v_i}$. If the vector $v_i$ is given, each $\varphi(v_i, e_j)$ is computable from $v_i$ in $O(b \log b)$ time by using a proper sparse representation\(^5\). Hence, the matrix $M_{v_i}$ is computable in $O(b^2 \cdot \log b) = O(B \log B)$. Using the same sparse representation, computing the product $M_{v_i} \cdot x$ takes an additional $O(B \log B)$ time for each $v_i$. This stage requires $O(B \log B \cdot n)$ computation time and $O(B^{1/2} \cdot n)$ communication units.

### 6.B.3.2 Minimum Bandwidth Repair

In what follows we show that it is possible to repair a node failure by communicating a single field element from $b - 1$ nodes. For functional repair no further computations are needed while for exact repair an additional $O(B^2)$ algorithm should be applied by the newcomer.

#### Algorithm 4

Let $v_j$ be the failed node, let $\{u_i\}_{i \in [b]}$ be a basis of $\mathbb{F}_q^b$, and $v_j = \sum_{i \in [b]} \gamma_i u_i$. The newcomer picks any $t \in [b]$ such that $\gamma_t \neq 0$, and downloads the element $\varphi(v_j, u_i) \cdot x$ from each of the nodes $\{u_i\}_{i \in [b]\setminus\{t\}}$.

Notice that for each $i \in [b]$, $\varphi(v, u_i) \in P_{<u_i>}$ by Lemma 49, and thus Algorithm [1] is well defined.

#### Lemma 52

By using the information received from Algorithm [1], the newcomer may restore the information from the failed node $v_j$ by using $O(B^2)$ field operations. For functional repair, no further computations are needed.

**Proof.** By Lemma [50], for any proper $t$ picked by the newcomer, the set $\{\varphi(v, u_i)\}_{i \in [b]\setminus\{t\}}$ is a basis for the subspace $P_{<v>}$. Hence, for functional repair no further computations are required by the newcomer. For exact repair the newcomer needs to perform Gaussian-like process on a matrix of size $(b - 1) \times B$. By Lemma [48] this process requires the same $O(b^2)$ row operations performed during a Gaussian elimination of a $(b - 1) \times b$ matrix. However, these row operations are being performed on rows of length $B$, and hence this Gaussian elimination requires $O(b^2 \cdot B) = O(B^3)$ field operations.

\[\blacksquare\]

#### Corollary 21

Using Algorithm [1], it is possible to add a new node that was not initially in the DSS (see Section [6.B.3.1]).

\(^5\)e.g., a sparse representation of $x = (\gamma_1, \ldots, \gamma_B)$ is $\{(j, \gamma_j)\}_{j \gamma_j \neq 0}$. This representation clearly requires $O(w_H(x) \cdot \log B) = O(w_H(x) \cdot \log b)$ space, where $w_H(x)$ is the Hamming weight of $x$. 

123

Technion - Computer Science Department - Ph.D. Thesis PHD-2017-01 - 2017
6.B.3.3 Local Repair

It is often required that a failed node will be repairable from as few other active nodes as possible. It is clear that without replication of nodes, a minimum of two active nodes is necessary for such a repair. In the following we present an alternative repairing approach that may achieve this minimum. The possibility of achieving this minimum depends on the specific assignment of vectors to the nodes. This assignment will be discussed in detail in Subsection 6.B.4.2.

**Algorithm 5** Let $v_j$ be the failed node and let $\{u_1, \ldots, u_\ell\}$ be a set of active linearly independent nodes such that $v_j \in \langle u_1, \ldots, u_\ell \rangle$. For all $t \in [\ell]$, the newcomer $\nu$ downloads the entire vector $M_{u_t} \cdot x$ from $u_t$.

**Lemma 53** By using the information received from Algorithm 2, the newcomer $\nu$ may restore the information of the failed node $v_j$ in $O(\ell^2 \cdot b)$ field operations.

**Proof.** Since $v_j \in \langle u_1, \ldots, u_\ell \rangle$, it follows that $v_j = \sum_{t=1}^\ell \gamma_t u_t$ for some $\gamma_1, \ldots, \gamma_\ell \in \mathbb{F}_q$. By the definition of the matrices $\{M_{u_1}, \ldots, M_{u_\ell}\}$, for all $t \in [\ell]$, the elements

$$\{\varphi(u_t, e_i) \cdot x\}_{i \in [b] \setminus \{r(v_j)\}}$$

are downloaded by $\nu$. Therefore, by Definition 17 and Lemma 48, for all $t \in [\ell]$ and all $w \in \mathbb{F}_b^q$, the newcomer may compute the element $\varphi(u_t, w) \cdot x$.

The newcomer computes the coefficients $\gamma_1, \ldots, \gamma_\ell$, e.g. by performing Gaussian elimination on the matrix whose rows are $\{u_1, \ldots, u_\ell, v_j\}$, a process requiring $O(\ell^2 \cdot b)$ field operation. Having these coefficients the newcomer performs

$$\sum_{t=1}^\ell \gamma_t \varphi(u_t, e_i) \cdot x = \varphi \left( \sum_{t=1}^\ell \gamma_t u_t, e_i \right) \cdot x = \varphi(v_j, e_i) \cdot x.$$

for all $i \in [b] \setminus \{r(v_j)\}$ in $O(\ell \cdot b)$ operations, and reassembles the vector $M_{v_j} \cdot x$. Overall, Algorithm 2 requires performing $O(\ell^2 \cdot b)$ field operations and communicating $\ell \cdot (b - 1)$ field elements. ■
Corollary 22 Let $v_j$ be a failed node. If $\ell$ is the smallest integer such $v_j$ is in the linear span of $\ell$ other active nodes, then the locality of repairing $v_j$ is $\ell$.

6.B.3.4 Parallel Repair

Consider the scenario of multiple simultaneous node failures. Obviously, under $t$ failures, if the conditions of Algorithm 2 are satisfied, then it is possible to execute $t$ sequential instances of the repair algorithm. We show that this could be improved in a certain special case. This is a simple consequence of Lemma 41.

Lemma 54 If $\{v_{i1}, \ldots, v_{it}\}$ is a set of failed nodes and $\{v_{j1}, \ldots, v_{js}\}$ is a set of active linearly independent nodes, of the remaining nodes, such that

$$\{v_{i1}, \ldots, v_{it}\} \subseteq \langle v_{j1}, \ldots, v_{js} \rangle,$$

then it is possible to repair all failures by communicating $s \cdot (b - 1)$ field elements.

Proof. Assume that a third party $\Psi$ is managing the repair process of all $t$ nodes simultaneously. $\Psi$ may download the entire content of all nodes $\{v_{j1}, \ldots, v_{js}\}$, and compute the set $\{\varphi(v_{im}, e_\ell) \cdot x\}_{\ell=1}^b$ for each $m \in [t]$ using Algorithm 2.

The complexity of Lemma 54 remains $t$ times the complexity of Algorithm 2. However, the amount of communication is the same as in a single instance of Algorithm 5. It is evident that this algorithm requires good locality. An assignment of vectors to nodes that achieves locality is discussed in Subsection 6.B.4.2.

6.B.3.5 Reconstruction

This subsection presents two reconstruction algorithms for two different models of communication. In Algorithm 3 which follows, the DC accesses $b - 1$ active nodes and downloads $b^2 - 3b + 3$ field elements overall. Algorithm 7 which follows, uses the additional assumption that the nodes participating in the reconstruction know the identities of one another (e.g., by broadcast, shared memory, or by acknowledgement from the DC), and
guarantees reconstruction by communicating $B$ field elements. This is the minimum communication that guarantees a complete reconstruction of $x$.

**Algorithm 6** Let $\{u_1, \ldots, u_{b-1}\}$ be any set of active linearly independent nodes. Let $u_b$ be a vector, not necessarily affiliated with an active node, which completes $\{u_1, \ldots, u_{b-1}\}$ to a basis of $\mathbb{F}_q^b$. For each $j \in [b-2]$, the DC downloads the vector $M_{u_j} \cdot x$ from $u_j$. In addition, the DC downloads the element $\varphi(u_{b-1}, u_b) \cdot x$ from $u_{b-1}$. The DC assembles the vector $w \in \mathbb{F}_q^b$ such that $w_{\{i,j\}} = \varphi(u_i, u_j) \cdot x$, and the $B \times B$ matrix $A$ whose rows are $\{\varphi(u_i, u_j)\}_{i \neq j}$. The vector $x$ is then reconstructed by solving the linear system of equations $Ax = w$.

**Lemma 55** The matrix $A$ in Algorithm 6 has full rank. In particular, the DC may extract $x$ by performing $O(B^3)$ field operations and downloading $(b-2)(b-1) + 1 = b^2 - 3b + 3$ field elements.

**Proof.** We first show that all entries of the vector $w$ are computable by the DC. If $i \leq b-2$, then by Lemma 49, the entry $\varphi(u_i, u_j) \cdot x$ can be computed from the information downloaded from $u_i$. If $i, j > b-2$, and hence, $\{i, j\} = \{b-1, b\}$, the corresponding entry was downloaded from $u_{b-1}$.

By Lemma 51, the matrix $A$ is invertible, and thus the reconstruction of $x$ is possible. Computing the rows of $A$ requires $O(B^2)$ operations, and solving a $B \times B$ linear system of equations requires additional $O(B^3)$ operations.

Assuming that every node participating in the reconstruction algorithm knows the identity of all other participating nodes, it is possible to reduce the communication to merely $|x| = B$ field elements from $b-1$ nodes, $\frac{b}{2}$ elements from each node if $b$ is even and either $\frac{b-1}{2}$ elements or $\frac{b+1}{2}$ elements if $b$ is odd. As mentioned earlier, this is the minimum possible communication since no outer code is used. The following matrix, whose construction is deferred to Appendix A, will be used in Algorithm 7.

**Definition 18** Let $N$ be a binary $b \times b$ matrix such that

1. For all $i \in [b]$, $N_{i,b} = 0$.

$^a$The entries of the vector $w \in \mathbb{F}_q^b$ are identified by all 2-subsets of $[b]$ according to the lexicographic order.
A2. For all \( j \in [b - 1] \), \( N_{b,j} = 1 \).

A3. For all \( i \in [b] \), \( N_{i,i} = 0 \).

A4. For all \( i, j \in [b], i \neq j \), \( N_{i,j} \neq N_{j,i} \).

A5. If \( b \) is even then for all \( i \in [b - 1] \) the Hamming weight of the \( i \)th column is \( \frac{b^2}{2} \).

A6. If \( b \) is odd then for all \( i \in [b - 1] \) the Hamming weight of the \( i \)th column is either \( \frac{b - 1}{2} \) or \( \frac{b + 1}{2} \) and the total Hamming weight of \( N \) is \( \binom{b}{2} \).

Example 5 The following matrices satisfy the requirements of Definition 18 for \( b = 6 \) and \( b = 5 \):

\[
\begin{pmatrix}
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
\end{pmatrix}
\quad \begin{pmatrix}
0 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 \\
\end{pmatrix}
\]

Algorithm 7 Let \( \{u_1, \ldots, u_b\} \) be any set of active linearly independent nodes and let \( N \) be the matrix from definition 18. For all \( i \in [b - 1] \), the DC downloads from node \( u_i \) all elements \( \varphi(u_i, u_j) \cdot x \) such that \( N_{j,i} = 1 \). The DC assembles the vector \( w \in \mathbb{F}_q^B \) such that \( w_{\{i,j\}} = \varphi(u_i, u_j) \cdot x \), and a \( B \times B \) matrix \( A \) whose rows are the vectors \( \{\varphi(u_i, u_j)\}_{i \neq j} \). The vector \( x \) is then reconstructed by solving the linear system of equations \( Ax = w \).

Lemma 56 Using the information received from Algorithm 7, the DC may construct the vector \( w \). In addition, the matrix \( A \) from Algorithm 7 has full rank, and hence \( x \) is reconstructible performing \( O(B^3) \) field operations and communicating \( B \) field elements.

Proof. By A4 of Definition 18 it is evident that for all \( i, j \in [b], i \neq j \), the DC receives the element \( \varphi(u_i, u_j) \cdot x \) exactly once. By Lemma 51 the matrix \( A \) is invertible, and thus the reconstruction of \( x \) is possible.
6.B.3.6 Modification

A useful property of a DSS is being able to update a small fraction of $x$ without having to initialize the entire system. The linear nature of our code and the absence of an outer code allows these modifications to be done efficiently. In particular, the complexity of the process is a function of the Hamming distance $d_H(x, y)$, where $y$ is the modification of the vector $x$. In an MDS based distributed storage systems a change of a single bit of $x$ usually requires changing a large portion of the data. Therefore, one more advantage of our system is revealed.

**Lemma 57** If $x \in \mathbb{F}_q^B$ is stored in the system, it is possible to update the system to contain $y \in \mathbb{F}_q^B$ by communicating $(\log B + \log q) \cdot d_H(x, y) \cdot n$ bits.

**Proof.** Each node receives a list $\{(\delta_i, \ell_i)\}_{i=1}^{d_H(x, y)}$, where $\delta_i \in \mathbb{F}_q$ and $\ell_i \in [B]$. The list indicates the values of the nonzero entries of the vector $y - x$. Each node $v$, holding the vector $M_v \cdot x$ (see Section 6.B.3.1) may assemble the matrix $M_v$ and compute:

$$M_v \cdot x + M_v \cdot (y - x) = M_v \cdot y.$$ 

Communicating the list $\{(\delta_i, \ell_i)\}_{i=1}^{d_H(x, y)}$ to all the $n$ nodes clearly requires $(\log B + \log q) \cdot d_H(x, y) \cdot n$ bits. ■

6.B.4 Assignment of Vectors

In Section 6.B.3 we proved that the performance of the detailed algorithms strongly relies on the chosen vectors $v_1, \ldots, v_n$. Since both repair and reconstruction algorithms require linearly independent nodes, it follows that the assigned set of vectors should contain a basis to $\mathbb{F}_q^b$ even after multiple failures.

Choosing $n = \left[\frac{b}{1}\right]_q$ and assigning all possible normalized vectors would suffice for repairing exponentially many failures. However, using $\left[\frac{b}{1}\right]_q = \Theta(q^b)$ storage nodes to store a file of size $B = \Theta(b^2)$ is unnecessary, as will be shown in the sequel. Furthermore, expecting exponentially many failures is nonrealistic.
In order to achieve reasonable failure resilience using a reasonable number of nodes, it suffices to consider the case \( n = O(b) \). Subsection 6.B.4.1 discusses an assignment of vectors compatible with Algorithm 1 presented in Subsection 6.A.3.1. An assignment compatible with Algorithms 2 of Subsection 6.A.3.2 and also for the algorithm of Subsection 6.A.3.3 is presented in Subsection 6.B.4.2.

**Definition 19** For \( t \in \mathbb{N} \) a set \( S \subseteq \mathbb{F}_q^b \) is called a \( t \)-resilient spanning set if every \( t \)-subset \( T \) of \( S \) satisfies \( \langle S \setminus T \rangle = \mathbb{F}_q^b \).

**Observation 3** If \( S \) is a \( t \)-resilient spanning set then by using \( |S| \) storage nodes assigned with the vectors in \( S \) (see Subsection 6.B.3.1), it is possible to repair and reconstruct in the presence of up to \( t \) simultaneous node failures.

**Example 6** The following set is a 2-resilient spanning set in \( \mathbb{F}_2^7 \):

\[
\begin{align*}
1 & 0 0 0 0 0 0 \\
0 & 1 0 0 0 0 0 \\
0 & 0 1 0 0 0 0 \\
0 & 0 0 1 0 0 0 \\
0 & 0 0 0 1 0 0 \\
0 & 0 0 0 0 1 0 \\
0 & 0 0 0 0 0 1 \\
1 & 1 1 1 1 1 1 \\
1 & 1 1 1 0 0 0 \\
1 & 1 0 0 1 1 0 \\
1 & 0 1 0 1 0 1
\end{align*}
\]

**6.B.4.1 Minimum Bandwidth Assignment**

In what follows we present a construction of a set of vectors \( \{v_1, \ldots, v_n\} \) compatible with Algorithm 1 achieving \( d \beta = \alpha \). This construction will rely on a well-known property of a generator matrix of a linear code.

**Lemma 58** If \( C \) is a linear block code over \( \mathbb{F}_q \) with length \( n \), dimension \( b \), distance \( \delta \), and generator matrix \( G \) (that is, \( C \triangleq \{xG| x \in \mathbb{F}_q^b\} \)), then the columns of \( G \) are a \((\delta - 1)\)-resilient spanning set.
Proof. Let \( S \subseteq \mathbb{F}_q^b \) be a set of \( n - \delta + 1 \) columns of \( G \) and let \( G' \in \mathbb{F}_q^{b \times (n-\delta+1)} \) be the matrix which consists of only these columns. We show that \( \dim \langle G' \rangle = b \). Assume for contradiction that there exists two different linear combinations of rows of \( G' \) that yield the same row vector, that is

\[
\sum_{i=1}^{b} \gamma_i G'_i = \sum_{i=1}^{b} \delta_j G'_j,
\]

where \( G'_i \) denotes the \( i \)th row of \( G' \) and \( \gamma_i, \delta_i \in \mathbb{F}_q \) for all \( i \in [b] \). Consider the codewords

\[
c_1 \triangleq \sum_{i=1}^{b} \gamma_i G_i, \quad c_2 \triangleq \sum_{j=1}^{b} \delta_j G_j,
\]

where \( G_i \) is the \( i \)th row of \( G \). Clearly, \( c_1 \) and \( c_2 \) are codewords of \( C \). However, they share \( n - \delta + 1 \) identical entries, which implies \( d_H(c_1, c_2) \leq \delta - 1 \), a contradiction to the minimum distance of \( C \). Therefore, there are \( q^b \) different vectors in \( \langle G' \rangle \), and \( \text{rank}(G') = \dim \langle G' \rangle = b \). Since the column and row ranks are equal, it follows that the set \( S \) spans \( \mathbb{F}_q^b \).

Surprisingly, the inverse of Lemma 46 is also true, as stated in the next lemma.

Lemma 59 Let \( S \subseteq \mathbb{F}_q^b \) be an assignment of vectors to nodes in some DSS which is resilient to \( \delta - 1 \) node failures by using the algorithms described in Section 6.B.3. If \( G \) is the matrix whose columns are the elements of \( S \) and \( C \triangleq \{ xG \mid x \in \mathbb{F}_q^b \} \) then \( C \) is a linear code of minimum Hamming distance \( \delta \).

Corollary 23 If \( C \) is a linear block code over \( \mathbb{F}_q \) with length \( n \), dimension \( b \), distance \( \delta \), and generator matrix \( G \), then by assigning the columns of \( G \) to nodes in the DSS, under any \( \delta - 1 \) node failures, there exists a set of active linearly independent nodes from which repair and reconstruction are possible.

In any linear code, the so called Singleton bound states that \( b \leq n - \delta + 1 \), and hence a set \( A \) of \( n - \delta + 1 \) nodes is not necessarily a basis to \( \mathbb{F}_q^b \). Therefore, the actual set of nodes required for repair and reconstruction is some \( b \)-subset.
of $A$. Notice that although this $b$-subset is guaranteed to exist by Lemma 46, not every $b$-subset of $A$ is necessarily a basis. However, when the code $C$ satisfies the MDS property (that is, $b = n - \delta + 1$) we get the following simple corollary.

**Corollary 24** If $C$ is an MDS code over $\mathbb{F}_q$ with length $n$, dimension $b$, distance $\delta$, and generator matrix $G$, then by assigning the columns of $G$ to nodes in the DSS, any set of $b$ active nodes allows repair and reconstruction.

Notice that Corollary 19 holds for any $q$, whereas Corollary 20 requires the existence of an MDS codes, and hence $q \geq n$ is required.

**Example 7** Let $C$ be a binary Justesen code [17] of length $O(b)$, dimension $b$, and minimum Hamming distance $\delta b$. We get that the corresponding code (see Section 6.B.3) uses $O(b)$ storage nodes while being able to recover from any $\delta b$ simultaneous node failures. In addition, the code uses the binary field. This choice admits the following parameters: $q = 2$, $B = \left(\frac{b}{2}\right)$, $n = O(B^{1/2})$, $d = k = \alpha = b - 1 = O(B^{1/2})$, and $\beta = 1$.

### 6.B.4.2 Minimum Locality Assignment

Algorithm 2 in Subsection 6.A.3.2 may possibly achieve the optimal locality. It is evident from Lemma 41 that in order to get good locality, the set $\{u_1, \ldots, u_\ell\}$ from Algorithm 2 is required to be small. However, this requirement conflicts with the requirements of Algorithms 1, 3, and 7, since they all involve large linearly independent sets.

In this subsection we show that by choosing some basis of $\mathbb{F}_b^q$, partitioning it to equally sized subsets and taking the linear span of each subset, some locality is achievable. The resulting failure resilience will grow with the field size. Thus, this technique will be particularly useful in large fields.

**Definition 20** Let $c$ be a positive integer such that $c$ divides $b$, and let $A \triangleq \{v_1, \ldots, v_b\}$ be a basis of $\mathbb{F}_q^b$. Partition $A$ into $\frac{b}{c}$ equally sized subsets $A_i \triangleq \{v_{ic + 1}, \ldots, v_{(i+1)c}\}$ for $i \in \{0, \ldots, \frac{b}{c} - 1\}$. Let $V_i \subseteq \mathbb{F}_q^b$ be a set of $\left[\frac{c}{1}\right]_q$ representatives for the 1-subspaces of $\langle A_i \rangle$. Finally, let $V \triangleq \bigcup_{i=1}^{b/c} V_i$.

**Lemma 60** The set $V$ from Definition 20 is a $(q^{c-1} - 1)$-resilient spanning set (see Definition 14). Furthermore, assigning $V$ to nodes in a DSS allows repairing any node failure using at most $c$ active nodes.
Proof. Since
\[ q^{c-1} - 1 = \left[ \frac{c}{1} \right]_q - \left[ \frac{c-1}{1} \right]_q - 1 < \left[ \frac{c}{1} \right]_q - \left[ \frac{c-1}{1} \right]_q, \]
it follows that after any set of at most \( q^{c-1} - 1 \) node failures, the set of remaining active nodes in any \( V_i \) is not contained in any \( (c-1) \)-subspace of \( \langle A_i \rangle \). Therefore, any \( V_i \) still contains a basis for \( \langle A_i \rangle \). Since \( \langle A_1 \rangle \oplus \cdots \oplus \langle A_{b/c} \rangle = \mathbb{F}_q^b \), it follows that \( V \) is \( (q^{c-1} - 1) \)-resilient spanning set.

Let \( v_j \) be a failed node and let \( V_t \) be the set containing it. We prove that \( v_j \) is repairable using at most \( c \) other nodes in the presence of at most \( q^{c-1} - 1 \) failures. We have shown that after \( q^{c-1} - 1 \) failures, the remaining active nodes in any given \( V_t \) contain a basis of \( \langle A_t \rangle \). Let \( \{u_1, \ldots, u_c\} \subseteq \langle A_t \rangle \) be such a basis in \( V_t \). It follows that \( v_j \in \langle u_1, \ldots, u_c \rangle \), and hence \( v_j \) is repairable by accessing at most \( c \) nodes by Lemma [I].

This construction requires \( \frac{b}{c} \cdot \left[ \frac{c}{1} \right]_q \) nodes and allows locality of \( c \) in the presence of up to \( q^{c-1} - 1 \) failures. For simple comparison, the trivial replication code with \( \frac{b}{c} \cdot \left[ \frac{c}{1} \right]_q \) nodes allows locality of 1 in the presence of up to \( \frac{1}{c} \cdot \left[ \frac{c}{1} \right]_q - 1 \) failures. We note that
\[
\frac{q^{c-1} - 1}{\frac{1}{c} \cdot \left[ \frac{c}{1} \right]_q - 1} \xrightarrow{q \to \infty} c,
\]
and in particular for \( c = 2 \),
\[
\frac{q^{2-1} - 1}{\frac{1}{2} \cdot \left[ \frac{2}{1} \right]_q - 1} = 2.
\]
Therefore, this code outperforms the trivial one by approximately a factor of \( c \) for large field size, while providing low locality. In particular, a minimal locality of 2 is achievable for any \( q \).

6.B.5 Previous Work

The code which was used as an inspiration to this paper is found in [12, Example 3.2]. It uses the subspace interpretation mentioned in Section 6.B.1.
**Construction 5** [13] Example 3.2] Let $x \in \mathbb{F}_q^B$, where $B = \binom{b}{2}$ for some $b \in \mathbb{N}$. Identify the coordinates of $\mathbb{F}_q^B$ with 2-subsets of $[b]$, and define the following set of subspaces

$$U_i = \langle e_{\{i,j\}} \rangle_{j \neq i}.$$ 

If the subspaces $U_i, i \in [b]$ are assigned to nodes, it is possible to store $x$ in $b$ nodes and support a single node failure. Reconstruction is possible by communicating all data from $b - 1$ nodes.

Our work has the same storage and same repair bandwidth as Construction 5, but it poses several advantages over it, such as repairing in the presence of a larger number of simultaneous failures and good locality. In particular, Example 7 in our paper uses $O(b)$ storage nodes while being able to repair $\delta b$ simultaneous node failures for some constant $\delta$.

Besides [12], equidistant subspace codes were also observed to be useful in [1]. However, no general non-trivial construction was made, and repair/reconstruction algorithms were not discussed. Trivial equidistant codes were used for a DSS in [11]. This code is also known as a spread, that is, all cyclic shifts of some subfield of $\mathbb{F}_q^n$. Although a spread provides a good locality, it is not clear how to use it in a DSS whose number of nodes is small.

Finally, a code with identical parameters may be achieved by using the Product-Matrix MBR code construction [11, Section IV]. In this construction, the file $x$ is turned into a symmetric matrix $M$, and each node stores $\psi_i^t M$, where $\psi_i$ is a row in a Vandermonde of Cauchy matrix $\Psi$. By setting $d = k = b - 1$ in this construction, we receive a code whose parameters are identical to those presented in Corollary 20. Although it is not mentioned in [9], one may replace the matrix $\Psi$ in any generator matrix of a linear block code over any finite field and receive a code with the same parameters, as well as the same repair and reconstruction capabilities, as the code mentioned in Corollary 19. In addition, the local repair algorithm mentioned in Section 6.A.3.2 is also possible, since the linearity mentioned in Lemma 48 is also evident in their work. However, it is not clear if reducing the reconstruction bandwidth from $(b - 1)^2$ (to $b^2 - 3b + 3$ in Algorithm 3 or to $B$ in Algorithm 7) is possible.

---

7A cyclic shift of a subfield $\mathbb{F}_{q^k}$ is a set of the form $\{\gamma v \mid v \in \mathbb{F}_{q^k}\}$ for some $\gamma \in \mathbb{F}_{q^n} \setminus \{0\}$. 

133
Bibliography


Appendix A

Two constructions of a matrix satisfying the requirements of Definition 18 are given, Construction 6 for even \( b \) and Construction 7 for odd \( b \). It is easily verified that these two constructions satisfy the requirement of Definition 18.

**Construction 6** Let \( b \) be an even integer. Define \( N \in \mathbb{F}_2^{b \times b} \) as follows. For all \( i \in [b] \) let \( N_{i,b} = 0 \) and for all \( i \in [b-1] \) let \( N_{b,i} = 1 \). The remaining \((b-1) \times (b-1)\) upper left submatrix is defined as follows. The first row is the \( b-1 \) bit vector \( 0^{b/2}1^{b/2-1} \); that is, \( \frac{b}{2} \) zeros followed by \( \frac{b}{2}-1 \) ones. The rest of the rows are all cyclic shifts of it (see Example 5 for the case \( b = 6 \)).

**Lemma 61** The matrix given by Construction 6 satisfies the requirements of Definition 18.

**Proof.** Conditions A1, A2, and A3 obviously hold. We prove conditions A4 and A5.

To prove Condition A4, let \( N' \) be the upper left \((b-1) \times (b-1)\) submatrix of \( N \) and let \( N'' \) be the \( (\frac{b}{2}-1) \times (\frac{b}{2}-1) \) upper right submatrix of \( N' \). Notice that \( N'' \) is an upper triangular matrix with 1 in entry \( i,j \) for every \( j \geq i \). In addition, \( N' \) has 0 in all entries above its main diagonal except those in \( N' \). Similarly, Let \( N''' \) be the \( (\frac{b}{2}-1) \times (\frac{b}{2}-1) \) bottom left submatrix of \( N' \). It may easily be seen to have 0 in entry \( i,j \) for every \( i \leq j \), and \( N' \) has 1 in all entries below the main diagonal except the 0 entries of \( N''' \).

To prove Condition A5 observe that since all cyclic shifts of the first row appear, each column contains every 1 from the first row exactly once, and has an additional 1 in the bottom entry. Hence, the Hamming weight of any column \( i \) where \( i \in [b-1] \) is \( (\frac{b}{2}-1)+1 = \frac{b}{2} \).

A similar construction can be done for an odd \( b \).

**Construction 7** Let \( b \) be an odd integer. Define \( N \in \mathbb{F}_2^{b \times b} \) as follows. For all \( i \in [b] \) let \( N_{i,b} = 0 \) and for all \( i \in [b-1] \) let \( N_{b,i} = 1 \). The remaining \((b-1) \times (b-1)\) upper left submatrix is defined as follows. The first row is the \( b-1 \) bit vector \( 0^{(b+1)/2}1^{(b-3)/2} \); that is, \( \frac{b+1}{2} \) zeros followed by \( \frac{b-3}{2} \) ones. The rest of the rows are all cyclic shifts of it. In addition, set the sub diagonal entries \((1,\frac{b-1}{2}+1),(2,\frac{b-1}{2}+2),\ldots,(\frac{b-1}{2},b-1)\) to 1 (see Example 5 for the case \( b = 5 \)).
Lemma 62  The matrix given by Construction 7 satisfies the requirements of Definition 18.
Chapter 7

Construction of High Rate Minimum Storage Regenerating Codes over Small Fields

Netanel Raviv, Natalia Silberstein, and Tuvi Etzion

7.A Conference Version

Abstract

This paper presents a new construction of high-rate minimum storage regenerating codes. In addition to a minimum storage in a node, these codes have the following two important properties: first, given storage $\ell$ in each node, the entire stored data can be recovered from any $2\log_2 \ell$ (any $3\log_3 \ell$) nodes for two parities (for three parities, respectively); second, a helper node accesses the minimum number of its symbols for repair of a failed systematic node (access-optimality). The goal of this paper is to provide a construction of such optimal codes over the smallest possible finite fields. The generator matrix of these codes is based on perfect matchings of complete graphs and hypergraphs, and on a rational canonical form of matrices. For two parities, the field size is reduced by a factor of two for access-optimal codes compared.
to previous constructions. For three parities, the field size is \(6 \log_3 \ell + 1\) (or \(3 \log_3 \ell + 1\) for fields with characteristic 2), where only non-explicit constructions with exponential field size (in \(\log_3 \ell\)) were known so far.

### 7.A.1 Introduction

Regenerating codes are a family of erasure codes proposed by Dimakis et al. [4] to store data in distributed storage systems (DSSs) in order to reduce the amount of data (repair bandwidth) downloaded during repair of a failed node. An \((n, k, \ell, d, \beta, B)_q\) regenerating code \(C\), for \(k \leq d \leq n - 1\), \(\beta \leq \ell\), is used to store a file of size \(B\) in a DSS across a network of \(n\) nodes, where each node of the system stores \(\ell\) symbols from \(\mathbb{F}_q\), such that the stored file can be recovered by downloading the data from any set of \(k\) nodes. When a single node fails, a newcomer node which substitutes the failed node, contacts any set of \(d\) nodes, called helper nodes, and downloads \(\beta\) symbols from each helper node to reconstruct the data stored in the failed node. This process is called a node repair process and the parameter \(d\) is called the repair degree. This paper deals with exact repair, where the newcomer node contains exactly the same data as the failed node.

Based on a min-cut analysis of the information flow graph which represents a DSS, Dimakis et al. [4] established a tradeoff between the number \(\ell\) of stored symbols in a node and the repair bandwidth \(\beta d\). Two classes of codes that achieve the two extreme points of this tradeoff are known as minimum storage regenerating (MSR) codes and minimum bandwidth regenerating (MBR) codes. The parameters \((\ell, \beta d)\) for MSR and MBR codes are given by \(\left(\frac{B}{k}, \frac{B d}{k(d-k+1)}\right)\) and \(\left(\frac{2Bd}{2kd-k^2+k}, \frac{2Bd}{2kd-k^2+k}\right)\), respectively [4].

Constructions of exact MSR and MBR codes (codes which support exact repair) can be found in [9, 10, 11, 12, 14, 16, 17] and references therein. MSR codes are in particular MDS array codes [2, 3].

In this paper we focus on exact MSR codes which provide minimum repair bandwidth of systematic nodes and have additional properties listed below.

1. **Maximum repair degree** \(d = n - 1\): this enables to minimize the repair bandwidth among all MSR codes. Such MSR codes satisfy \((\ell, \beta d) = \left(\frac{B}{k}, \frac{B(n-1)}{k(n-k)}\right)\) [4].

139
2. **High rate** $\frac{k}{n}$: in particular the number of parity nodes $r = n - k$ is $r = 2$ or $r = 3$ (see e.g. [9, 16, 17] for previously known constructions of such MSR codes).

3. **Optimal access**: the number of symbols accessed in a helper node is minimal and equals to the number $\beta$ of symbols transmitted during node repair. (See [1] [15] for bounds and constructions of access-optimal codes.)

4. **Optimal sub-packetization factor**: for an $(n, k, \ell, d, \beta, B)_q$ regenerating code, the number of stored symbols $\ell$ in a node is also called the *sub-packetization* factor of the code. Low-rate ($\frac{k}{n} \leq \frac{1}{2}$, i.e., $\frac{n}{r} \geq \frac{1}{2}$) MSR codes with $2k - 2 \leq d \leq n - 1$, where $\ell$ is linear in $r$ were constructed in [10]. However, in the known high-rate ($\frac{k}{n} > \frac{1}{2}$) MSR codes $\ell$ is exponential in $k$ [9, 17]. Moreover, it was proved in [15] that for an access-optimal code, given a fixed sub-packetization factor $\ell$ and $r$ parity nodes, the largest number $k$ of systematic nodes is

$$k = r \log_r \ell, \quad (7.1)$$

i.e., the required sub-packetization factor $\ell$ is $r^k$.

5. **Small finite field**: construction of access-optimal MSR codes with $r = 2$ and optimal sub-packetization $2^k$ over a finite field of size $2\log_2 \ell + 1$ was presented in [17]. More precisely, this construction provides a code with a larger number $k' = 3 \log_2 \ell$ of systematic nodes out of which $k = 2 \log_2 \ell$ have the optimal access property. Hence the shortened code with $k$ systematic nodes is an access-optimal code. For general $r$, codes with sub-packetization factor $\ell = r^m$ and $k = rm$, over $\mathbb{F}_q$, where $q \geq k^{r-1} r^{m-1} + 1$ were presented in [17] and codes for any $q \geq \binom{n}{k} r^{m+1}$ were presented in [1].

In addition, a construction for $r = 2$ which achieves $k = 2 \log_2 \ell$ and requires $q \geq \log_2 \ell + 1$ was presented in [7]. This construction requires $q$ to be even, and while it does not provide the optimal access property, it provides a different property called *optimal update*.\(^1\)

\(^{1}\)DSS codes which satisfy the property that any change in the original data requires minimal updates at the storage nodes are called optimal update codes.
We propose a construction of access-optimal MSR codes with optimal sub-packetization factor $\ell = r^m$, $k = rm$, for $r = 2$ and $r = 3$, over any finite field $\mathbb{F}_q$ such that $q \geq m + 1$ and $q \geq 6m + 1$, respectively. Moreover, for $r = 3$, if $q$ is a power of 2 then the field size can be reduced to $q \geq 3m + 1$. In addition, we present a construction of a longer code which is not access-optimal, with $r = 3$, $k = 4m$, and $q = \Theta(m)$. The comparison of the results presented in this paper with some previously known MSR codes can be found in Tables 7.1 and 7.2 for two and three parity nodes, respectively.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$k$</th>
<th>access optimality</th>
<th>field size $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2^m$</td>
<td>$2m$</td>
<td>✓ for $m$ nodes</td>
<td>$m + 1$</td>
</tr>
<tr>
<td>$2^m$</td>
<td>$3m$</td>
<td>✓ for $2m$ nodes</td>
<td>even $m + 1$</td>
</tr>
<tr>
<td>$2^m$</td>
<td>$2m$</td>
<td>✓</td>
<td>$2m + 1$</td>
</tr>
<tr>
<td>$2^m$</td>
<td>$m + 1$</td>
<td>✓</td>
<td>$2m + 1$</td>
</tr>
</tbody>
</table>

Table 7.1: Comparison of our code with some previously known MSR codes with two parities.

<table>
<thead>
<tr>
<th>$\ell$</th>
<th>$k$</th>
<th>access optimality</th>
<th>field size $q$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3^m$</td>
<td>$3m$</td>
<td>✓ for $3m$ nodes</td>
<td>even $3m + 1$; odd $6m + 1$</td>
</tr>
<tr>
<td>$3^m$</td>
<td>$4m$</td>
<td>✓</td>
<td>$\Theta(m)$</td>
</tr>
<tr>
<td>$3^m$</td>
<td>$3m$</td>
<td>✓</td>
<td>$m^{2(m+1)} + 1$</td>
</tr>
<tr>
<td>$3^m$</td>
<td>$m + 1$</td>
<td>✓</td>
<td>$(\frac{3m+3}{3})^{3m+1}$</td>
</tr>
</tbody>
</table>

Table 7.2: Comparison of our code with some previously known MSR codes with three parities.

The main technique used in this paper is based on a construction of a
maximal set of pairs of vector subspaces and matrices which satisfy the so-called subspace condition [5, 6, 15, 17]. The existence of such a set provides the sufficient condition for existence of MSR codes with optimal sub-packetization factor [15, 17]. When in addition the subspaces of this sequence are generated by the unit vectors (binary vectors of weight one) the related MSR code has the optimal access property. Therefore, in order to construct an access-optimal MSR code with optimal sub-packetization factor over a small finite field, we focus on the algebraic problem of construction of a maximum size set of subspaces and matrices which is defined over a small field and which satisfies the subspace condition, with the subspaces spanned by the unit vectors.

We associate the unit vectors with the vertices of a complete hypergraph. Our construction of vector subspaces for the set is based on perfect matchings in such a hypergraph. The construction of matrices for the set is based on both the perfect matchings and on the specific rational canonical form. In particular, we use the fact that the rational canonical form of any matrix (even non-diagonalizable) exists over any field.

The rest of the paper is organized as follows. In Section 7.A.2, the underlying algebraic problem is described. In Section 7.A.3, the outline for all our constructions is explained. Our constructions of codes with two and three parity nodes, are presented in Sections 7.A.4 and 7.A.5, respectively.

Due to space limitations, some proofs are only sketched and some are omitted. The interested reader can find the proofs in the full version of this paper [13].

7.A.2 The Subspace Condition

In many real-world applications, a DSS is required to have a systematic part, i.e., certain nodes in the system should store an uncoded part of the data. Such nodes are called systematic nodes, and they allow instant access to their stored data. An efficient repair algorithm for a failed systematic node is vital. In this paper, we devise an MSR code which allows a minimum repair bandwidth for any failed systematic node.

Construction of such MSR codes was previously studied in [5, 6, 15, 17], where it was shown to be equivalent to a purely algebraic problem called the subspace condition, which is briefly described in this section.
As mentioned in the Introduction, MSR codes are in particular MDS array codes. In an MSR code with \( k \) systematic nodes, \( r \) parity nodes, and sub-packetization \( \ell \), a file \( f \in \mathbb{F}_q^{k \ell} \) is partitioned into \( k \) parts of length \( \ell \) each, denoted by \( f = (C_1, \ldots, C_k) \). The file \( f \) is multiplied by a \( k\ell \times (k + r)\ell \) generator block matrix of the form
\[
\begin{pmatrix}
I & A_{1,0} & A_{1,1} & \cdots & A_{1,r-1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
I & A_{k,0} & A_{k,1} & \cdots & A_{k,r-1}
\end{pmatrix}
\]
(7.2)
where \( I \) is the \( \ell \times \ell \) identity matrix, and the \( A_{i,j} \)'s are invertible matrices, which satisfy a certain set of properties [17]. For simplicity, we assume that \( A_{i,j} = A^T_{j,i} \) for some \( A_1, \ldots, A_k \) that will be defined in the sequel.

The resulting codeword is partitioned into \( k + r \) columns of length \( \ell \) each, denoted by \((C_1, \ldots, C_k, C_{k+1}, \ldots, C_{k+r})\), where for all \( j \in [r] \triangleq \{1, \ldots, r\} \),
\[
C_{k+j} = \sum_{i=1}^{k} A_{j-1} C_i.
\]
Each column \( C_i \) is stored in a different storage node, where the first \( k \) nodes are the systematic ones and the remaining \( r \) nodes are parity nodes. The sufficient condition for the minimum repair bandwidth is as follows.

**Definition 21** *(The Subspace Condition, [15, Section II]*) Let \( \ell \) and \( r \) be integers such that \( r \) divides \( \ell \). A set of pairs \( \{(A_i, S_i)\}_{i=1}^{k} \) where for all \( i \), \( A_i \) is an invertible \( \ell \times \ell \) matrix and \( S_i \) is an \( \ell/r \) dimensional subspace of \( \mathbb{F}_q^\ell \), satisfies the subspace condition if the following three properties hold.

**The independence property:** for each \( i \in [k] \),
\[
S_i + S_i A_i + S_i A_i^2 + \ldots + S_i A_i^{r-1} = \mathbb{F}_q^\ell.
\]

**The invariance property:** for all \( i, j \in [k], i \neq j \), \( S_i A_j = S_i \).

**The nonsingular property:** Every square block submatrix of the follow-
ing block matrix is invertible.

\[
\begin{pmatrix}
I & A_1 & \cdots & A_{r-1} \\
\vdots & \vdots & \ddots & \vdots \\
I & A_k & \cdots & A_{r-1}
\end{pmatrix}
\]

If a subspace \( S \) satisfies the invariance property for a matrix \( A \), then \( S \) is an invariant subspace of \( A \). If a subspace \( S' \) satisfies the independence property for \( A \), then \( S' \) is an independent subspace of \( A \). Notice that the nonsingular property must hold for the code to be an MDS array code [2, 3], regardless of any applications in distributed storage.

**Theorem 33** [17] If the set \( \{(A_i, S_i)\}_{i=1}^k \) satisfies the subspace condition for given \( \ell \) and \( r \), then the code whose generator matrix is given in (7.2) is an MSR code which allows a minimum repair bandwidth for any systematic node.

The subspaces \( \{S_i\}_{i=1}^k \) in Theorem 33 are used in the repair process. To repair a systematic node \( i \), the remaining nodes project their data on \( S_i \) and send it to the newcomer node.

In order to compute the projections on the subspace \( S_j \), each of the remaining nodes must access a certain amount of its stored symbols, and clearly, at least \( \ell/r \) symbols must be accessed. A code in which this minimum is attained is called an access-optimal code [7, 16, 17]. It can be shown that a code is access-optimal if and only if each subspace \( S_j \) has a basis which consists only of unit vectors [15, Section V].

A set of the form \( \{(A_i, S_i)\}_{i=1}^k \) is called an \((A, S)\)-set. Since the subspace condition is sufficient for construction of such MSR codes, this paper will focus solely on the construction of \((A, S)\)-sets which satisfy it.

### 7.A.3 Our Techniques

Our constructions rely on the properties of some matrix \( A \), to which the matrices in our \((A, S)\)-set are similar using certain change matrices\(^2\). These

---

\(^2\)Matrices \( A \) and \( B \) which satisfy \( B = P^{-1}AP \) are called similar matrices. The matrix \( P \) is called a change-of-basis matrix, or a change matrix in short.
change matrices are defined according to a set of matchings in the complete $r$-uniform hypergraph on $\ell$ vertices $K^r_{\ell}$. In this subsection the matrix $A$ is described, its properties are discussed, and the use of matchings for the definition of the change matrices is explained.

The matrix $A$ and the change matrices will be described with respect to a construction with $r$ parities, for a general $r$. In the following sections, the case of $r = 2$ will be discussed in detail, and the case of $r = 3$ will be briefly mentioned.

For a given number of parities $r$ and an integer $m$, the matrix $A$ is an $r^m \times r^m$ block diagonal matrix whose constituent blocks are the $r \times r$ companion matrix $C$ \cite[Ch. IX, S. 6]{8} of the polynomial $x^r - 1$. That is,

$$C \triangleq \begin{pmatrix} 0 & \cdots & 0 & 1 \\ I_{r-1} & 0 & \cdots & 0 \\ \vdots \end{pmatrix}, \quad \text{and} \quad A \triangleq \begin{pmatrix} C & \cdots \\ & \ddots \\ & & C \end{pmatrix}. \quad (7.3)$$

Since it is desirable that $A$ will have as many eigenspaces as possible, we operate over a field $\mathbb{F}_q$, where $r | q - 1$. This assumption about $q$ provides the existence of all roots of unity $1, \gamma_1, \ldots, \gamma_{r-1}$ of order $r$ in the field $\mathbb{F}_q$ (using the well-known Sylow theorems \cite[Section XII.5]{8}). It is readily verified that the eigenvalues of $A$ are $1, \gamma_1, \ldots, \gamma_{r-1} \in \mathbb{F}_q$, since they are the roots of the minimal polynomial $x^r - 1$ of $A$. We note that for the special case of $r = 2$ (Section 7.4.A), we use an additional technique which allows to operate with any $q \geq m + 1$, without requiring that $2 | q - 1$.

The matrices in our construction are similar to the matrix $A$. The change matrices which induce the similarity are defined using perfect colored matchings in the complete $r$-uniform hypergraph. Although the specific choice of these change matrices varies from one construction to another, the general idea behind the use of matchings is roughly identical, and will be explained in the remainder of this subsection.

**Definition 22** A perfect colored matching (matching, in short) is a perfect matching in the $r$-uniform hypergraph, whose edges are colored in $r$ colors such that no edge contains two nodes of the same color.

We denote a matching by $Z = (Z^{(0)}, \ldots, Z^{(r-1)})$, where each $Z^{(i)}$ is an
ordered color set, and if \( Z^{(i)} = (z^{(i)}_0, \ldots, z^{(i)}_{\ell/r-1}) \) for each \( i \in \{0, \ldots, r-1\} \),
then the edges of \( Z \) are
\[
\left\{ \{z^0_j, z^1_j, \ldots, z^{(r-1)}_j\} \right\}_{j=0}^{\ell/r-1}.
\]

For example, for \( r = 2 \), a matching is denoted by \( Z = (Z, Z') \) (we use \( Z \) and \( Z' \) instead of \( Z^{(0)} \) and \( Z^{(1)} \) for convenience), where \( Z = (z_0, \ldots, z_{\ell/2-1}) \), \( Z' = (z'_0, \ldots, z'_{\ell/2-1}) \), and the edges of \( Z \) are \( \{\{z_i, z'_i\}\}_{i=0}^{\ell/2-1} \).

Each matching will be used to construct \( r \) (or \( r + 1 \) in Section 7.A.5) change matrices for the \((A,S)\)-set. Each \( \ell \times \ell \) change matrix is constructed using constituent \( r \times \ell \) sub-matrices. Each such submatrix is a function of a single edge in the matching. That is, if the matching is \( Z = (Z^{(0)}, \ldots, Z^{(r-1)}) \),
then \( r \) matrices in the \((A,S)\)-set are constructed as \( A_i = P_i^{-1}AP_i \), where
\[
P_i = \begin{pmatrix}
\text{An } r \times \ell \text{ submatrix based on } \\
\{z^0_0, z^1_0, \ldots, z^{(r-1)}_0\}
\vdots
\text{An } r \times \ell \text{ submatrix based on } \\
\{z^0_{\ell/r-1}, z^1_{\ell/r-1}, \ldots, z^{(r-1)}_{\ell/r-1}\}
\end{pmatrix}.
\tag{7.4}
\]

The vertices of the \( r \)-uniform hypergraph \( \mathcal{K}_r^\ell \) are identified with the \( \ell \) unit vectors \( e_0, \ldots, e_{\ell-1} \). In all subsequent constructions, the subspaces in the \((A,S)\)-set are defined using the color sets from the matchings, i.e., if \( Z = (Z^{(0)}, \ldots, Z^{(r-1)}) \) is a matching, then we define \( r + 1 \) subspaces of dimension \( \ell/r \) as follows. For all \( i \in \{0, \ldots, r-1\} \),
\[
S_{Z^{(i)}} \triangleq \left\langle Z^{(i)} \right\rangle,
\quad S_{Z^*} \triangleq \left\langle \{z^0_i + \cdots + z^{(r-1)}_i\}_{i=0}^{\ell/r-1} \right\rangle.
\tag{7.5}
\]

That is, each subspace \( S_{Z^{(i)}} \) is the span of the color set \( Z^{(i)} \), and the additional subspace \( S_{Z^*} \) is the span of the sums of each edge in \( Z \). To enlarge the \((A,S)\)-set, different matchings can be used, as long as they satisfy the
following simple condition.

**Definition 23** Two matchings \( \mathcal{X} = (X^{(0)}, \ldots, X^{(r-1)}) \) and \( \mathcal{Y} = (Y^{(0)}, \ldots, Y^{(r-1)}) \) satisfy the pairing condition if any edge in \( \mathcal{X} \) is monochromatic in \( \mathcal{Y} \), and vice versa.

In the sequel we use a large set of matchings in which every two matchings satisfy the pairing condition. To satisfy the nonsingular property (Definition 21), each matrix is multiplied by a properly chosen field constant. The constructions of the \((A, S)\)-sets, which follow the general outline described in this subsection, are discussed in detail in the following sections.

### 7.A.4 MSR Codes with Two Parities

#### 7.A.4.1 Two Parities Code from One Matching

Recall that the vertices of the complete graph \( K_\ell \) are identified with all unit vectors \( e_0, \ldots, e_{\ell-1} \) of length \( \ell = 2^m \), for some integer \( m \), and a matching \( Z = (Z, Z') \) is a set of \( \ell/2 \) vertex-disjoint edges of \( K_\ell \). Such a matching will provide an \((A, S)\)-set of size 2, satisfying the subspace condition. The construction of this \((A, S)\)-set also relies on the following \( \ell \times \ell \) matrices, which resemble the matrix in \((7.3)\). For \( \lambda \in \mathbb{F}_q^* \), consider the following two \( \ell/2 \times \ell/2 \) matrices

\[
A^+(\lambda) \triangleq \begin{pmatrix} 0 & \lambda & \cdots & 0 & 0 \\ \lambda & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & \lambda \\ 0 & 0 & \cdots & \lambda & 0 \end{pmatrix}, \quad A^-(-\lambda) \triangleq -A^+(\lambda),
\]

and let \( A(\lambda) \) be the following \( \ell \times \ell \) block diagonal matrix

\[
A(\lambda) \triangleq \begin{pmatrix} A^+(\lambda) & 0 \\ 0 & A^-(\lambda) \end{pmatrix}.
\]  \((7.6)\)

Over fields with characteristic two, the matrix \( A(\lambda) \) is non-diagonalizable. To the best of our knowledge, this constitutes the first construction of
an \((A, S)\)-set satisfying the subspace condition whose matrices are non-diagonalizable. Notice that the multiplication of a vector \(v\) by the matrix \(A(\lambda)\) switches between entries \(2t\) and \(2t + 1\) of \(v\) for all \(t \in \{0, \ldots, \ell/2 - 1\}\), and multiplies all entries by either \(\lambda\) or \(-\lambda\) according to \(t \leq \ell/4 - 1\) or \(t > \ell/4 - 1\).

Given a matching \(Z = (Z, Z')\), it can be easily verified that the following two matrices are invertible. Recall that \(z_i, z'_i\) are vertices in the complete graph, which are identified by unit vectors of length \(\ell\).

\[
P_Z \triangleq \begin{pmatrix} z_0 & z'_0 & \cdots & z_{\ell/2 - 1} \\ z'_0 - z_0 & \vdots & \vdots & \vdots \\ z_{\ell/2 - 1} & \vdots & \vdots & \vdots \\ z_{\ell/2 - 1} - z'_0 \end{pmatrix}, \quad P_{Z'} \triangleq \begin{pmatrix} z'_0 & z_0 & \cdots & z'_{\ell/2 - 1} \\ z_0 + z'_0 & \vdots & \vdots & \vdots \\ z_{\ell/2 - 1} & \vdots & \vdots & \vdots \\ z_{\ell/2 - 1} + z'_0 \end{pmatrix}
\]

**Definition 24** Given a matching \(Z = (Z, Z')\), let

\[
A_Z(\lambda) \triangleq P^{-1}_Z \cdot A(\lambda) \cdot P_Z, \quad S_Z \triangleq \langle Z \rangle = \{z_i\}_{i=0}^{\ell/2-1}
\]

\[
A_{Z'}(\lambda) \triangleq P^{-1}_Z \cdot A(\lambda) \cdot P_{Z'}, \quad S_{Z'} \triangleq \langle Z' \rangle = \{z'_i\}_{i=0}^{\ell/2-1}
\]

The matching \(Z\) provides an \((A, S)\)-set of size two as follows.

**Lemma 63** If \(Z = (Z, Z')\) is a matching, then \(\{(A_Z(\lambda), S_Z), (A_{Z'}(\lambda), S_{Z'})\}\) satisfies the subspace condition.

From Lemma 63 it is evident that any pair \((Z, \lambda)\) of a matching \(Z = (Z, Z')\) and a nonzero field element \(\lambda\) provides an \((A, S)\)-set of size two. In Section 7.A.4.2 we discuss the required relation between two such pairs \((X, \lambda_x), (Y, \lambda_y)\) that allow the corresponding \((A, S)\)-sets to be united without compromising the subspace condition.

### 7.A.4.2 Two Parities Code from Two Matchings

A set \(\{(X_i, \lambda_i)\}_{i=1}^{t}\) such that any two pairs satisfy the conditions given below, will provide an \((A, S)\)-set of size 2\(t\). In the sequel we provide such a set of size \(m\) over \(\mathbb{F}_q\), for any \(m \in \mathbb{N}\) and any \(q \geq m + 1\). This set will yield an \((A, S)\)-set of size 2\(m\) for \(q \geq m + 1\), which consists of matrices of size \(2^m \times 2^m\).
Lemma 64 If $\mathcal{X} = (X, X')$, $\mathcal{Y} = (Y, Y')$ are matchings and $\lambda_x, \lambda_y$ are nonzero field elements such that

A1. $\lambda_x \neq \lambda_y$.

A2. $\mathcal{X}$ and $\mathcal{Y}$ satisfy the pairing condition.

A3. If $\lambda_x = -\lambda_y$, then for all $i \in \{0, \ldots, \ell/2 - 1\}$, if $(x_i, x'_i) = (y_j, y_t)$ then

$$i \leq \ell/4 - 1, \quad j \leq \ell/4 - 1, \quad \text{and} \quad t > \ell/4 - 1,$$

and if $(x_i, x'_i) = (y'_j, y'_t)$ then

$$i > \ell/4 - 1, \quad j \leq \ell/4 - 1, \quad \text{and} \quad t > \ell/4 - 1.$$

then the $(\mathcal{A}, \mathcal{S})$-set

$$\{(A_X(\lambda_x), S_X), (A_X(\lambda_x), S_X'), (A_Y(\lambda_y), S_Y), (A_Y(\lambda_y), S_Y')\}$$

satisfies the subspace condition.

By Lemma 64 we have that two matchings $\mathcal{X}$, $\mathcal{Y}$ and two corresponding field elements $\lambda_x, \lambda_y$ that meet the requirements A1-A3, provide an $(\mathcal{A}, \mathcal{S})$-set of size four. Therefore, a construction of a large set of pairs $(X_i, \lambda_i)$, such that any two pairs satisfy A1-A3, is required for a construction of a large $(\mathcal{A}, \mathcal{S})$-set which satisfies the subspace condition.

7.A.4.3 Construction of Matchings for Two Parities

In the sequel we construct a set $\{(X_i, \lambda_i)\}_{i=0}^{m-1}$ whose elements satisfy the requirements of Lemma 64 in pairs. For convenience we identify vertex $e_i$ of $K_\ell$ with the integer $i$ in its binary representation. We will use the following standard notion of a Boolean cube.

Definition 25 Given a sequence of distinct integers $i_1, \ldots, i_k$ in $\{0, \ldots, m - 1\}$ and a sequence of Boolean values $b_1, \ldots, b_k$, the Boolean cube $C(\{(i_j, b_j)\}_{j=1}^k)$ is the set of all $m$-bit vectors over $\{0, 1\}$ that have $b_j$ in entry $i_j$ for all
That is,
\[ C(\{(i_j, b_j)\}_{j=1}^k) \triangleq \{ x \in \{0,1\}^m \mid \text{for all } j \in [k], \ x_{i_j} = b_j \} . \]

For convenience, we consider the elements in such a Boolean cube as ordered according to the lexicographic order, that is, we consider a Boolean cube as a sequence rather than a set.

**Definition 26** For any \( m \in \mathbb{N} \), define \( m \) matchings \( \{X_i = (X_i, X'_i)\}_{i=0}^{m-1} \) as follows

\[ X_{2t} : \begin{cases} 
X_{2t} = C(\{(2t,0), (2t+1,0)\}) \circ C(\{(2t,0), (2t+1,1)\}) \\
X'_{2t} = C(\{(2t,1), (2t+1,0)\}) \circ C(\{(2t,1), (2t+1,1)\})
\end{cases} \]

\[ X_{2t+1} : \begin{cases} 
X_{2t+1} = C(\{(2t,0), (2t+1,0)\}) \circ C(\{(2t,1), (2t+1,0)\}) \\
X'_{2t+1} = C(\{(2t,0), (2t+1,1)\}) \circ C(\{(2t,1), (2t+1,1)\})
\end{cases} \]

where \( t \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor - 1 \} \), and \( \circ \) indicates the concatenation of sequences.

If \( m \) is odd, we add the matching

\[ X_{m-1} : \begin{cases} 
X_{m-1} = C(\{(m-1,0)\}) \\
X'_{m-1} = C(\{(m-1,1)\})
\end{cases} \]

**Lemma 65** Every two distinct matchings \( X_i, X_j \) from Definition 26 satisfy the pairing condition.

We now turn to choose a proper nonzero field element for every matching from Definition 26. This choice must comply with requirements A1 and A3 of Lemma 64. Note that if \( q \) is even, then A3 follows from A1. Hence, in these fields the choice of field elements is straightforward.

**Lemma 66** If \( q \geq m + 1 \) is a power of two, then by arbitrarily assigning pairwise distinct elements from \( \mathbb{F}_q^* \) to the \( m \) matchings from Definition 26, the resulting \( (A, S) \)-set satisfies A1-A3 from Lemma 64.
If \( q \) is odd, more care is needed for the mapping of nonzero field elements to the matchings. We do this by assigning the field elements \( \lambda \) and \( -\lambda \) to two adjacent matchings \( X_{2t}, X_{2t+1} \).

**Lemma 67** If \( q \geq m + 1 \) is a power of an odd prime, then by arbitrarily assigning pairwise distinct elements from \( \mathbb{F}_q^* \) to the \( m \) matchings from Definition 26, such that \( X_{2t}, X_{2t+1} \) are mapped to additive inverses \( \lambda, -\lambda \) for every \( t \in \{0, \ldots, \left\lfloor \frac{m}{2} \right\rfloor - 1 \} \), the resulting \((A,S)\)-set satisfies A1-A3 from Lemma 64.

The main construction of this section is summarized in the following theorem.

**Theorem 34** If \( m \) is a positive integer and \( q \geq m + 1 \) is a prime power, then there exists an explicitly defined \((A,S)\)-set \( C \) of size \( 2m \) and \( 2^m \times 2^m \) matrices over \( \mathbb{F}_q \), which satisfies the subspace condition.

### 7.A.5 MSR Codes with Three Parities

In this section we present a brief outline of our two constructions for MSR codes with three parities. The first construction provides access-optimal codes with \( k = 3m \), and requires \( q \geq 6m + 1 \) (or \( q \geq 3m + 1 \) if \( q \) is even). The second construction provides codes which are not access-optimal, with \( k = 4m \), and requires \( q = \Theta(m) \). Both constructions involve a plethora of technical details, given in [13].

Both constructions, which follow the outline described in Section 7.A.3, make use of the following matrix (see (7.3)).

\[
C \triangleq \begin{pmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0
\end{pmatrix}
\quad A \triangleq \begin{pmatrix}
C & & \\
& & \\
& C
\end{pmatrix}
\]

The matrices in the \((A,S)\)-set are obtained via similarity to \( A \). The change matrices which induce this similarity are based on perfect colored matchings in the complete 3-uniform hypergraph, whose vertices are identified by all unit vectors in \( \mathbb{F}_q^\ell \). Each constituent \( 3 \times \ell \) block of the change matrices (7.4)
is a function of a hyperedge in a matching, and certain field constants which are chosen according to field characteristics.

Similar to Definition 24, a single matching provides three pairs \((A_i, S_i)\) in the first construction and four pairs in the second, with the addition of the subspace \(S_{Z^*}\) in the latter \((7.5)\).

In order to use more than one matching, the pairing condition must hold (Definition 23). The proof that the nonsingular property still holds when uniting codes from different matchings, uses the remarkable fact that matrices which correspond to different matchings are simultaneously diagonalizable, and hence they commute. These techniques give rise to the following theorems.

**Theorem 35** If \(m\) is a positive integer, and \(q\) is a prime power such that

1. if \(q\) is odd, then \(3|q - 1\) and \(q \geq 6m + 1\),

2. if \(q\) is even, then \(3|q - 1\) and \(q \geq 3m + 1\),

then there exists an explicitly defined \((A, S)\)-set \(C_1\) of size \(3m\) and \(3^m \times 3^m\) matrices over \(\mathbb{F}_q\), which satisfies the subspace condition.

In what follows, the polynomial \(Q\) is a constant degree polynomial which arises in the proof of the theorem.

**Theorem 36** If \(q > \max\{42m, \deg Q\} + 1\), then there exists an explicitly defined \((A, S)\)-set \(C_2\) of size \(4m\) and \(3^m \times 3^m\) matrices over \(\mathbb{F}_q\), which satisfies the subspace condition.

**Acknowledgments**

This research was supported in part by the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant no. 10/12.

The work of Natalia Silberstein was also supported in part by the Fine fellowship.

The work of Netanel Raviv was also supported in part by the IBM Ph.D. fellowship. It is a part of his Ph.D. thesis performed at the Technion.
Bibliography


7.B Unpublished Full Version

Abstract

A novel technique for construction of minimum storage regenerating (MSR) codes is presented. Based on this technique, three explicit constructions of MSR codes are given. The first two constructions provide access-optimal MSR codes, with two and three parities, respectively, which attain the sub-packetization bound for access-optimal codes. The third construction provides longer MSR codes with three parities (i.e., codes with larger number of systematic nodes). This improvement is achieved at the expense of the access-optimality and the field size.

In addition to a minimum storage in a node, all three constructions allow the entire data to be recovered from a minimal number of storage nodes. That is, given storage \( \ell \) in each node, the entire stored data can be recovered from any \( 2 \log_2 \ell \) for two parities nodes, and either \( 3 \log_3 \ell \) or \( 4 \log_3 \ell \) for three parities. Second, in the first two constructions, a helper node accesses the minimum number of its symbols for repair of a failed node (access-optimality). The goal of this paper is to provide a construction of such optimal codes over the smallest possible finite fields. The generator matrix of these codes is based on perfect matchings of complete graphs and hypergraphs, and on a rational canonical form of matrices. For two parities, the field size is reduced by a factor of two for access-optimal codes compared to previous constructions. For three parities, in the first construction the field size is \( 6 \log_3 \ell + 1 \) (or \( 3 \log_3 \ell + 1 \) for fields with characteristic 2), and in the second construction the field size is larger, yet linear in \( \log_3 \ell \). Both constructions with three parities provide a significant improvement over existing previous works, since only non-explicit constructions with exponential field size (in \( \log_3 \ell \)) were known so far.

7.B.1 Introduction

Regenerating codes are a family of erasure codes proposed by Dimakis et al. [4] to store data in distributed storage systems (DSSs) in order to reduce the amount of data (repair bandwidth) downloaded during a repair of a failed node. An \((n, k, \ell, d, \beta, B)_q\) regenerating code \(C\), for \(k \leq d \leq n - 1\),
\( \beta \leq \ell \), is used to store a file of size \( B \) in a DSS across a network of \( n \) nodes, where each node of the system stores \( \ell \) symbols from \( \mathbb{F}_q \), a finite field with \( q \) elements, such that the stored file can be recovered by downloading the data from any set of \( k \) nodes. When a single node fails, a newcomer node which substitutes the failed node, contacts any set of \( d \) nodes, called helper nodes, and downloads \( \beta \) symbols from each helper node to reconstruct the data stored in the failed node. This process is called a node repair process and the parameter \( d \) is called the repair degree. There are two general methods of node repairs: functional repair and exact repair. Functional repair ensures that when a node repair process is completed, the system is equivalent to the original one, i.e., the stored file can be recovered from any \( k \) nodes. However, the newcomer node may contain a different data from what was stored in the failed node. Exact repair requires that the newcomer node will store exactly the same data as was stored in the failed node. Usually, exact repair is required for systematic nodes (nodes that contain the actual data), while the parity nodes can be functionally repaired.

Based on a min-cut analysis of the information flow graph which represents a DSS, Dimakis et al. \[4\] presented an upper bound on the size of a file that can be stored using a regenerating code under functional repairs,

\[
B \leq \sum_{i=1}^{k} \min\{(d - i + 1)\beta, \ell\}.
\]

Given the values of \( B, n, k, d \), this bound provides a tradeoff between the number \( \ell \) of stored symbols in a node and the repair bandwidth \( \beta d \). The extremal point on this tradeoff, where \( \ell \) is minimized is referred to as minimum storage regenerating (MSR) point, and a code that attains it, namely, a minimum storage regenerating (MSR) code satisfies \[4\]

\[
(\ell, \beta d) = \left( \frac{B}{k}, \frac{Bd}{k(d - k + 1)} \right).
\]

(7.7)

The other extremal point, where \( \beta \ell \) is minimized is referred to as minimum bandwidth regenerating (MBR) point, and a code that attains it, namely, a minimum bandwidth regenerating (MBR) code satisfies \[4\]

\[
(\ell, \beta d) = \left( \frac{2Bd}{2kd - k^2 + k}, \frac{2Bd}{2kd - k^2 + k} \right).
\]

Constructions of MBR codes which support exact repair can be found for
example in [11, 12, 13]. Constructions of MSR codes which support exact repair can be found in [10, 11, 12, 13, 14, 15, 16] and references therein. Note that MSR codes are in particular MDS array codes [2, 3].

In this paper we focus on exact MSR codes which provide minimum repair bandwidth of systematic nodes and have additional properties listed below.

1. **Maximum repair degree** $d = n - 1$: this enables to minimize the repair bandwidth among all MSR codes. Such MSR codes satisfy $(\ell, \beta d) = \left(\frac{B}{k}, \frac{B(n-1)}{k(n-k)}\right)$ [4].

2. **High rate** $\frac{k}{n}$: in particular the number of parity nodes $r = n - k$ is $r = 2$ or $r = 3$ (see e.g. [10, 16, 15] for previously known constructions of such MSR codes).

3. **Optimal access**: the number of symbols accessed in a helper node is minimal and equals to the number $\beta$ of symbols transmitted during node repair. (See [1, 17] for bounds and constructions of access-optimal codes.)

4. **Optimal sub-packetization factor**: for an $(n, k, \ell, d, \beta, B)_q$ regenerating code, the number of stored symbols $\ell$ in a node is also called the sub-packetization factor of the code. Low-rate $(\frac{k}{n} \leq \frac{1}{2}$, i.e., $\frac{r}{n} \geq \frac{1}{2}$) MSR codes with $d = n - 1$, where $\ell$ is linear in $r$ were constructed in [11]. However, in the known high-rate $(\frac{k}{n} > \frac{1}{2}$) MSR codes $\ell$ is exponential in $k$ [10, 15]. Moreover, it was proved in [17] that for an access-optimal code, given a fixed sub-packetization factor $\ell$ and $r$ parity nodes, the largest number $k$ of systematic nodes is

$$k = r \log_r \ell,$$

i.e., the required sub-packetization factor $\ell$ is $r^{\frac{k}{\ell}}$.  

5. **Small finite field**: construction of access-optimal MSR codes with $r = 2$ and optimal sub-packetization $2^{\frac{1}{2}}$ over a finite field of size $1 + 2 \log_2 \ell$ is presented in [15]. More precisely, this construction provides a code with a larger number $k' = 3 \log_2 \ell$ of systematic nodes out of which $k = 2 \log_2 \ell$ have the optimal access property. Hence the shortened code with $k$ systematic nodes is an access-optimal code.
For general $r$, codes with sub-packetization factor $\ell = r^m$ and $k = rm$, over $\mathbb{F}_q$, where $q \geq k^{r-1}r^{m-1} + 1$ were presented in [15] and codes for any $q \geq \binom{r}{k} r^{m+1}$ were presented in [1]. In addition, a construction for $r = 2$ which achieves $k = 2 \log_2 \ell$ and requires $q \geq \log_2 \ell + 1$ is presented in [7]. This construction requires $q$ to be even, and while it does not provide the optimal access property, it provides a different property called optimal update.

We propose a construction of access-optimal MSR codes with optimal sub-packetization factor $\ell = r^m$, $k = rm$, for $r = 2$ and $r = 3$, over any finite field $\mathbb{F}_q$ such that $q \geq m + 1$ and $q \geq 6m + 1$, respectively. Moreover, for $r = 3$, if $q$ is a power of 2 then the field size can be reduced to $q \geq 3m + 1$. In addition, we present a construction of a longer code which is not access-optimal, with $r = 3$, $k = 4m$, and $q = \Theta(m)$.

The comparison of the results presented in this paper with some previously known MSR codes can be found in the following tables.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\ell = r^m$</td>
<td>$2^m$</td>
<td>$2^m$</td>
<td>$2^m$</td>
</tr>
<tr>
<td>$k = rm$</td>
<td>$2m$</td>
<td>$2m$</td>
<td>$2m$</td>
</tr>
<tr>
<td>access optimality</td>
<td>✓ for $m$ nodes</td>
<td>for $2m$ nodes</td>
<td>✓</td>
</tr>
<tr>
<td>field size $q$</td>
<td>$m + 1$</td>
<td>even $m + 1$</td>
<td>$2m + 1$</td>
</tr>
</tbody>
</table>

Table 7.3: Comparison of our code with some previously known MSR codes with two parities.

7.B.1.1 Organization

The construction for $r = 2$ is given in Section 7.B.4 and for $r = 3$ in Section 7.B.5. Additional construction for $r = 3$, which does not have the access-optimal property, is given in Section 7.B.6. Section 7.B.2 and

---

3DSS codes which satisfy the property that any change in the original data requires minimal updates at the storage nodes are called optimal update codes.
Table 7.4: Comparison of our code with some previously known MSR codes with three parities.

Section 7.B.3 present the techniques and underlying ideas used in Sections 7.B.4, 7.B.5, and 7.B.6. Subsection 7.B.2.1 contains some necessary mathematical background, Subsection 7.B.2.2 describes the underlying algebraic problem, and Section 7.B.3 explains the outline for all constructions.

7.B.2 Preliminaries

7.B.2.1 Algebra of Matrices - Background and Notations

The constructions in Section 7.B.4 and Section 7.B.5 extensively use several standard linear-algebraic notions. For the sake of completeness, we include below a short introduction about these necessary notions. Some of the given background is not directly used in the constructions, but may assist the reader with understanding our techniques, and their underlying reasoning.

For a prime power \( q \), \( \mathbb{F}_q^* \) is the set \( \mathbb{F}_q \setminus \{0\} \), \( \mathbb{F}_q^\ell \) is a vector space of dimension \( \ell \) over \( \mathbb{F}_q \), which consists of vectors of length \( \ell \), and \( \mathbb{F}_q^{\ell \times \ell} \) is the set of all \( \ell \times \ell \) matrices with entries in \( \mathbb{F}_q \). It is widely known that a matrix \( M \in \mathbb{F}_q^{\ell \times \ell} \) admits (left) eigenvectors and eigenvalues \(^9\) \( \text{Section VII.7} \). If \( v \in \mathbb{F}_q^\ell \) and \( vM = \lambda v \) for some \( \lambda \in \mathbb{F}_q \), then \( v \) is called a (left) eigenvector for the eigenvalue \( \lambda \). The linear span of all eigenvectors for a certain eigenvalue

\(^4\)Unless otherwise stated, all multiplications of a vector \( v \) by a matrix \( M \) in this paper are from the left, i.e., \( vM \).
\( \lambda \) is a subspace of \( \mathbb{F}_q^{\ell} \), and it is called a (left) eigenspace of \( M \).

For a subspace \( S \) of \( \mathbb{F}_q^{\ell} \), let \( SM \triangleq \{ sM \mid s \in S \} \). The set \( SM \) is obviously a subspace of \( \mathbb{F}_q^{\ell} \), and if \( M \) is invertible then \( \dim S = \dim(SM) \).

A subspace \( S \) which satisfies \( SM = S \) is called a (left) invariant subspace of \( M \) [9, Section XI.4] (in short, \( S \) is \( M \) invariant). Clearly, an eigenspace of \( M \) is also an invariant subspace of \( M \), but not necessarily vice versa.

For a polynomial \( p(x) \in \mathbb{F}_q[x] \) such that \( p(x) = \sum_{i=0}^{\ell-1} p_i x^i \), let \( p(M) \triangleq \sum_{i=0}^{\ell-1} p_i M^i \). The characteristic polynomial \( c(x) \) of \( M \) is the determinant of \( M - xI \) [9, Section IX.5], where \( I \) is the \( \ell \times \ell \) identity matrix, i.e. \( c(M) = 0 \). Furthermore, there exists a unique monic polynomial \( m(x) \in \mathbb{F}_q[x] \), of minimum degree, such that \( m(M) = 0 \). The polynomial \( m(x) \), called the minimal polynomial of \( M \), divides the characteristic polynomial \( c(x) \) of \( M \), and its roots are the eigenvalues of \( M \).

If \( P \in \mathbb{F}_q^{\ell \times \ell} \) is an invertible matrix, then the matrices \( P^{-1}MP \) and \( M \) are called similar matrices, and the matrix \( P \) is called a change matrix, (or a change-of-basis matrix) [9, Section VII.7]. It is easily verified that if \( e_0, \ldots, e_{\ell-1} \) is the standard basis of \( \mathbb{F}_q^{\ell} \), and \( p_0, \ldots, p_{\ell-1} \) are the rows of \( P \), then \( P^{-1}MP \) acts on \( p_0, \ldots, p_{\ell-1} \) exactly as \( M \) acts on \( e_0, \ldots, e_{\ell-1} \). That is, if

\[
\left( \sum_{i=0}^{\ell-1} \mu_i e_i \right) M = \sum_{i=0}^{\ell-1} \delta_i e_i
\]

for some coefficients \( (\mu_i)_{i=0}^{\ell-1} \) and \( (\delta_i)_{i=1}^{\ell-1} \), then

\[
\left( \sum_{i=0}^{\ell-1} \mu_i p_i \right) \left( P^{-1}MP \right) = \sum_{i=0}^{\ell-1} \delta_i p_i.
\]

As a result of this fact, we have that similar matrices share the same eigenvalues, but not necessarily the same eigenvectors. In addition, similar matrices also share the same minimal polynomial [9, Section IX.7].

Determining matrix similarity is possible by converting given matrices to one of several canonical forms. One such canonical form, which does not always exist, is the diagonal form. If a matrix \( M \) is similar to a diagonal matrix then \( M \) is called a diagonalizable matrix. It is well known that \( M \) is diagonalizable if and only if there exists a basis of \( \mathbb{F}_q^{\ell} \) in which all vectors are eigenvectors of \( M \) [9, Section VII.8, Theorem 19], and two matrices with the same diagonal form are similar. Two diagonalizable matrices \( A \) and \( B \)
are called \emph{simultaneously diagonalizable} if there exists an invertible matrix \( P \) such that \( P^{-1}AP \) and \( P^{-1}BP \) are diagonal matrices. Equivalently, \( A \) and \( B \) are simultaneously diagonalizable if there exists a basis of \( \mathbb{F}_q^{\ell} \) whose vectors are eigenvectors of \( A \) and of \( B \). Since diagonal matrices commute, it is easy to prove that simultaneously diagonalizable matrices commute as well.

Determining matrix similarity for matrices which are not necessarily diagonalizable is a corollary of the so-called decomposition theorem \cite{9} Section XI.4, Theorem 8], one of the profoundest results in linear algebra. In order to state this theorem, we require the notions of a block diagonal matrix and a companion matrix. A block diagonal matrix is a block matrix (that is, a matrix which is interpreted as being partitioned to submatrices) in which the only non-zero sub-matrices are on the main diagonal. A companion matrix of a polynomial \( p(x) \) is a \( \deg p \times \deg p \) matrix consisting of 1’s in the main sub-diagonal, the additive inverses of the coefficients of \( p \) in the rightmost column, and 0 elsewhere.

The decomposition theorem states that any matrix \( M \) is similar to a block diagonal matrix, whose blocks are companion matrices of certain factors of the characteristic polynomial. The polynomials corresponding to these companion matrices may be ordered such that any polynomial is a multiple of the next, and the first one is the minimal polynomial of \( M \). This block diagonal matrix is called the \emph{rational canonical form} (rational form, in short) of \( M \), any matrix \( M' \) is similar to \( M \) if and only if the share the same rational form, which exists over any field.

In what follows, for a set of row vectors \( T \) we denote its \( \mathbb{F}_q \)-linear span by \( \langle T \rangle \) and for subspaces \( U \) and \( V \), let \( U + V \triangleq \{ u + v \mid u \in U, v \in V \} \). For a matrix \( M \) we denote its (left) image by \( \text{Im}(M) \triangleq \{ vM \mid v \in \mathbb{F}_q^{\ell} \} \) and its row span by \( \langle M \rangle \).

\section{7.B.2.2 The Subspace Condition}

Usually, a distributed storage system has a \emph{systematic part}, i.e. certain nodes in the system should store an unencoded part of the data. Such nodes are called \emph{systematic nodes}, and they allow instant access to their stored data. An efficient repair algorithm for a failed systematic node is vital. In this paper, we devise an MSR code which allows a minimum repair bandwidth
for a failed systematic node.

This problem was previously studied by [15, 6, 5, 17], where it was shown to be equivalent to a purely algebraic condition called the subspace condition\(^5\). In this subsection we describe this condition, and explain why codes which satisfy it provide minimum repair bandwidth for a failed systematic node. We refer the interested reader to [15] for a proof that the subspace condition is also necessary. A more general formulation of the subspace condition, which is irrelevant in our context, may also be found in [15].

In an MSR code with \( k \) systematic nodes, \( r \) parity nodes, sub-packetization \( \ell \), and maximum repair degree \( d = n - 1 \), a file \( f \in \mathbb{F}_q^{k\ell} \) is partitioned into \( k \) parts of length \( \ell \) each, denoted by \( f = (C_1, \ldots, C_k) \). The file \( f \) is multiplied by a \( k\ell \times (k + r)\ell \) generator block matrix of the form

\[
\begin{pmatrix}
I & A_{1,0} & A_{1,1} & \cdots & A_{1,r-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
I & A_{k,0} & A_{k,1} & \cdots & A_{k,r-1}
\end{pmatrix}
\] (7.10)

where \( I \) is the \( \ell \times \ell \) identity matrix, and the \( A_{i,j} \)'s are invertible matrices, which satisfy a certain set of properties [15]. In this paper \( A_{i,j} = A_i^j \) for some \( A_1, \ldots, A_k \) that will be defined in the sequel. That is, (7.10) simplifies to

\[
\begin{pmatrix}
I & I & A_1 & \cdots & A_1^{r-1} \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
I & I & A_k & \cdots & A_k^{r-1}
\end{pmatrix}
\] (7.11)

The resulting codeword is partitioned into \( k + r \) columns of length \( \ell \) each, denoted \( (C_1, \ldots, C_k, C_{k+1}, \ldots, C_{k+r}) \), where for all \( j \in [r] = \{1, \ldots, r\} \),

\[
C_{k+j} = \sum_{i=1}^{k} A_i^{j-1} C_i.
\]

Each column \( C_i \) is stored in a different storage node, where the first \( k \) nodes are the systematic ones and the remaining \( r \) nodes are called parity nodes.

Upon a failure of a systematic node \( m \in [k] \), storing \( C_m \), it is required to repair it by downloading a minimal amount of data. According to (7.7), [15, 17] used the term "subspace property".
since \( n = k + r \) and \( d = n - 1 \), we have that the minimum bandwidth \( \beta_d \) in this scenario is

\[
\beta_d = \frac{B}{k} \cdot \frac{d}{d-k+1} = \frac{k \ell}{k} \cdot \frac{k+r-1}{r} = \ell \frac{(k+r-1)}{r}.
\]

That is, each of the remaining \( k + r - 1 \) nodes should contribute \( 1/r \) of its stored data [4]. Sufficient conditions for this minimum repair bandwidth are as follows.

**Definition 27** *(The Subspace Condition, [17, Section II]*) Let \( \ell \) and \( r \) be integers such that \( r \) divides \( \ell \). A set of pairs \( \{(A_i, S_i)\}_{i=1}^k \), where for all \( i \), \( A_i \) is an invertible \( \ell \times \ell \) matrix and \( S_i \) is an \( \ell/r \)-subspace of \( \mathbb{F}_q^{\ell} \), satisfies the subspace condition if the following properties hold.

**The independence property:**

For each \( i \in [k] \),

\[
S_i + S_iA_i + S_iA_i^2 + \ldots + S_iA_i^{r-1} = \mathbb{F}_q^{\ell}.
\]

**The invariance property:**

For all \( i, j \in [k], i \neq j \), \( S_iA_j = S_i \).

**The nonsingular property:**

Every square block submatrix of the following block matrix is invertible.

\[
\begin{pmatrix}
I & A_1 & \cdots & A_1^{r-1} \\
\vdots & \vdots & \ddots & \vdots \\
I & A_k & \cdots & A_k^{r-1}
\end{pmatrix}
\]

(7.13)

If a subspace \( S \) satisfies the invariance property for a matrix \( A \), then \( S \) is an invariant subspace of \( A \) (see Section 7.B.2.1). If a subspace \( S' \) satisfies the independence property for \( A \), then \( S' \) is an independent subspace of \( A \). Notice that the nonsingular property must hold for the code to be an MDS array code [2, 3], regardless of any applications in distributed storage.

**Theorem 37** *(Ex)* If the set \( \{(A_i, S_i)\}_{i=1}^k \) satisfies the subspace condition for given \( \ell \) and \( r \), then the code whose generator matrix is given in (7.11) is an MSR code which allows a minimum repair bandwidth for any systematic node.

163
The subspaces \( \{S_i\}_{i=1}^k \) in this theorem are used in the repair process, and are often called *repair subspaces*. To repair a systematic node \( j \), the remaining nodes project their data on \( S_j \), i.e. multiply their data by some full rank matrix whose row span is \( S_j \), and send it to the newcomer. For additional details see [15, 6, 5, 17].

In order to compute the projections on the subspace \( S_j \), each of the remaining nodes must access a certain amount of its stored symbols, and clearly, at least \( \ell/r \) symbols must be accessed. A code in which this minimum is attained is called an access-optimal code [16, 15, 7]. It can be shown that a code is access-optimal if and only if each subspace \( S_j \) has a basis which consists of unit vectors only [17, Section V].

A set of the form \( \{(A_i, S_i)\}_{i=1}^k \) is called an \( (A, S) \)-set. Since the subspace condition is necessary and sufficient for construction of MSR codes, this paper will focus solely on the construction of \( (A, S) \)-sets which satisfy it.

### 7.B.3 Our Techniques

Our constructions rely on the properties of some matrix \( A \), to which the matrices in our \( (A, S) \)-set are similar using certain change matrices. These change matrices are defined according to a set of matchings in the complete \( r \)-uniform hypergraph on \( \ell \) vertices \( K_{r\ell} \). In this subsection the matrix \( A \) is described, its properties are discussed, and the use of matchings for the definition of the change matrices is explained.

The matrix \( A \) and the change matrices will be described with respect to a construction with \( r \) parities, for a general \( r \). In the following sections, the cases of \( r = 2 \) and \( r = 3 \) will be discussed in detail.

For a given number of parities \( r \) and an integer \( m \), the matrix \( A \) is an \( r^m \times r^m \) block diagonal matrix whose constituent blocks are the \( r \times r \) companion matrix of \( x^r - 1 \). That is,

\[
C \triangleq \begin{pmatrix}
0 & \cdots & 0 & 1 \\
& & & \\
0 & \ddots & \ddots & \\
& & \ddots & \\
I_r - 1 & & & \\
& & \ddots & \\
& & & 0
\end{pmatrix},
\]
where $I_{r-1}$ is the identity matrix of order $r$, and the matrix $A$ is

$$A \triangleq \begin{pmatrix} C \\ & \ddots \\ & & C \end{pmatrix}. \quad (7.14)$$

Since it is desirable that $A$ will have as many eigenspaces as possible, we operate over a field $\mathbb{F}_q$, where $r|q-1$. This assumption about $q$ provides the existence of all roots of unity $1, \gamma_1, \ldots, \gamma_{r-1}$ of order $r$ in the field $\mathbb{F}_q$ (using the well-known Sylow theorems [9, Section XII.5]). It is readily verified that the eigenvalues of $A$ are $1, \gamma_1, \ldots, \gamma_{r-1} \in \mathbb{F}_q$, since they are the roots of the minimal polynomial $x^r - 1$ of $A$. We note that for the special case of $r = 2$ (Section 7.B.4), we use an additional technique which allows to operate with any $q \geq m + 1$, without requiring that $2|q-1$.

In what follows we present the structure of the eigenspaces of $A$, and the eigenspaces of matrices which are similar to $A$.

**Lemma 68** The matrix $C \in \mathbb{F}_q^{r\times r}$ is a diagonalizable matrix whose set of linearly independent eigenvectors is $\{(1, \ldots, 1)\} \cup \{(1, \gamma_i, \gamma_i^2, \ldots, \gamma_i^{r-1})\}_{i=1}^{r-1}$. Furthermore, the subspace $S = \langle e_0 \rangle$ is an independent subspace of $C$, i.e. $S + SC + \ldots + SC^{r-1} = \mathbb{F}_q^r$.

**Proof.** It is readily verified that for all $i \in \{0, \ldots, r-1\}$, $e_iC = e_{i-1 \mod r}$. Hence, $(1, \ldots, 1)$ is an eigenvector for the eigenvalue 1. In addition, for any $i \in [r-1]$, we have that

$$(1, \gamma_i, \gamma_i^2, \ldots, \gamma_i^{r-1})C = (\gamma_i, \gamma_i^2, \ldots, \gamma_i^{r-1}, 1) = \gamma_i(1, \gamma_i, \gamma_i^2, \ldots, \gamma_i^{r-1}),$$

and thus the vector $(1, \gamma_i, \gamma_i^2, \ldots, \gamma_i^{r-1})$ is an eigenvector which corresponds to the eigenvalue $\gamma_i$. These eigenvectors form an $r \times r$ Vandermonde matrix [8, p. 270], and hence they are linearly independent and $C$ is diagonalizable. Since for all $i \in \{0, \ldots, r-1\}$, $e_iC = e_{i-1 \mod r}$, we also have that $S + SC + \ldots + SC^{r-1} = \mathbb{F}_q^r$. \[\blacksquare\]

The structure of the eigenspaces and eigenvalues of $A$ is a simple corollary of Lemma 68.

**Corollary 25** The matrix $A$ is diagonalizable, and the linearly independent eigenvectors of $A$ are as follows.

---

165
1. For the eigenvalue $1$, the $\ell/r$ linearly independent eigenvectors are

\[
(1,1,\ldots,1, \quad 0,0,\ldots,0, \quad \cdots,0,0,\ldots,0) \\
(0,0,\ldots,0, \quad 1,1,\ldots,1, \quad \cdots,0,0,\ldots,0) \\
\vdots \\
(0,0,\ldots,0, \quad 0,0,\ldots,0, \quad \cdots,1,1,\ldots,1)
\]

2. For the eigenvalue $\gamma_i$, $i \in [r-1]$, the $\ell/r$ linearly independent eigenvectors are

\[
(1,\gamma_i,\ldots,\gamma_i^{r-1}, \quad 0,0,\ldots,0, \quad \cdots,0,0,\ldots,0) \\
(0,0,\ldots,0, \quad 1,\gamma_i,\ldots,\gamma_i^{r-1}, \quad \cdots,0,0,\ldots,0) \\
\vdots \\
(0,0,\ldots,0, \quad 0,0,\ldots,0, \quad \cdots,1,\gamma_i,\ldots,\gamma_i^{r-1})
\]

In addition, if $S \triangleq \langle e_0, e_r, e_{2r}, \ldots, e_{\ell-r} \rangle$ then $S + SA + SA^2 + \ldots + SA^{r-1} = \mathbb{F}_q^\ell$.

The matrices in our construction are similar to the matrix $A$. The following lemma, that is based on (7.9), presents the eigenvalues and eigenvectors of a matrix which is similar to $A$.

**Lemma 69** If $P \in \mathbb{F}_q^{\ell \times \ell}$ is an invertible matrix whose rows are $p_0, \ldots, p_{\ell-1}$, and $B \triangleq P^{-1}AP$, then $B$ is diagonalizable, with the following eigenspaces,

A1. For the eigenvalue $1$, a basis of the eigenspace is

\[
\left\{ \sum_{j=0}^{r-1} p_{ir+j} \mid i \in \{0,\ldots,\ell/r - 1\} \right\}.
\]

A2. For the eigenvalue $\gamma_i$, a basis of the eigenspace is

\[
\left\{ \sum_{j=0}^{r-1} \gamma_i^j p_{ir+j} \mid i \in \{0,\ldots,\ell/r - 1\} \right\}.
\]

In addition, the subspace $T \triangleq \langle p_0, p_r, p_{2r}, \ldots, p_{\ell-r} \rangle$ satisfies $T + TB + TB^2 + \ldots + TB^{r-1} = \mathbb{F}_q^\ell$. 

166
**Proof.** Notice that the vectors in $A_1$ and $A_2$ are given by multiplying the eigenvectors of $A$ (see Corollary 25) by the matrix $P$. If $v$ is an eigenvector of $A$ which corresponds to an eigenvalue $\lambda$ then

$$(vP)B = (vP)P^{-1}AP = vAP = \lambda vP,$$

and hence $vP$ is an eigenvector of $B$.

Notice that the subspace $T$ may be written as

$$T = \langle e_0^P, e_r^P, \ldots, e_{\ell-r}^P \rangle = \langle e_0, e_r, \ldots, e_{\ell-r} \rangle P^2,$$

where $S = \langle e_0, e_r, e_2r, \ldots, e_{\ell-r} \rangle$. Hence, it follows from Corollary 25 that

$$T + TB + TB^2 + \ldots + TB^{r-1} = SP + SPB + SPB^2 + \ldots + SPB^{r-1}$$

$$= SP + SAP + SA^2P + \ldots + SA^{r-1}P$$

$$= (S + SA + SA^2 + \ldots + SA^{r-1})P$$

$$= \mathbb{F}_q^\ell P = \mathbb{F}_q^\ell.$$

The matrices in our construction are similar to the matrix $A$. The change matrices which induce the similarity are defined using perfect colored matchings in the complete $r$-uniform hypergraph. Although the specific choice of these change matrices varies from one construction to another, the general idea behind the use of matchings is roughly identical, and will be explained in the remainder of this subsection.

**Definition 28** A perfect colored matching (matching, in short) is a perfect matching in the $r$-uniform hypergraph, whose vertices are colored in $r$ colors, such that no edge contains two nodes of the same color (see Figure 7.1).

We denote a matching by $Z = (Z^{(0)}, \ldots, Z^{(r-1)})$, where each $Z^{(i)}$ is an ordered color set (i.e. a subset of all vertices which are colored in the same color), and if $Z^{(i)} = (z_0^{(i)}, \ldots, z_{\ell/r-1}^{(i)})$ for each $i \in \{0, \ldots, r-1\}$, then the edges of $Z$ are

$$\left\{ \left\{ z_j^{(0)}, z_j^{(1)}, \ldots, z_j^{(r-1)} \right\} \right\}_{j=0}^{\ell/r-1}.$$

For example, for $r = 2$, a matching is denoted by $Z = (Z, Z')$ (we use $Z$ and $Z'$ instead of $Z^{(0)}$ and $Z^{(1)}$ for convenience), where $Z = (z_0, \ldots, z_{\ell/2-1})$, $Z' = (z_0', \ldots, z'_{\ell/2-1})$, and the edges of $Z$ are $\{\{z_i, z'_i\}\}_{i=0}^{\ell/2-1}$.

Each matching will be used to construct $r$ (or $r + 1$ in Section 7.B.6) change matrices for the $(A, S)$-set. Each $\ell \times \ell$ change matrix is constructed
Figure 7.1: A perfect colored matching on the complete (2-uniform hyper)graph with four vertices. The edges of the matching appear in bold, and the nodes are colored in ♠, ♣ such that there is no monochromatic edge.

Using constituent $r \times \ell$ sub-matrices (7.15). Each such submatrix is a function of a single edge in the matching. That is, if the matching is $Z = (Z^{(0)}, \ldots, Z^{(r-1)})$, then $r$ matrices in the $(A, S)$-set are constructed as $A_i = P_i^{-1} A P_i$, where

$$P_i = \begin{pmatrix}
\text{An } r \times \ell \text{ submatrix based on } \\
\{z_0^{(0)}, z_0^{(1)}, \ldots, z_0^{(r-1)}\}
\end{pmatrix}
\begin{pmatrix}
\text{An } r \times \ell \text{ submatrix based on } \\
\{z_1^{(0)}, z_1^{(1)}, \ldots, z_1^{(r-1)}\}
\end{pmatrix}
\cdots
\begin{pmatrix}
\text{An } r \times \ell \text{ submatrix based on } \\
\{z_{\ell/r-1}^{(0)}, z_{\ell/r-1}^{(1)}, \ldots, z_{\ell/r-1}^{(r-1)}\}
\end{pmatrix}.
$$

(7.15)

The vertices of the $r$-uniform hypergraph $K_\ell^r$ are identified with the $\ell$
unit vectors $e_0, \ldots, e_{\ell-1}$. In all subsequent constructions, the subspaces in the $(\mathcal{A}, \mathcal{S})$-set are defined using the color sets from the matchings, i.e., if $\mathcal{Z} = (Z^{(0)}, \ldots, Z^{(r-1)})$ is a matching, then we define $r$ subspaces $(r+1$ in Section 7.B.6) of dimension $\ell/r$ as follows

$$\forall i \in \{0, \ldots, r-1\}, \quad S_{Z^{(i)}} \triangleq \langle Z^{(i)} \rangle,$$

$$S_{Z^*} \triangleq \left\langle \left\{ z_i^{(0)} + z_i^{(1)} + \cdots + z_i^{(r-1)} \right\}_{i=0}^{\ell/r-1} \right\rangle.$$

That is, each subspace $S_{Z^{(i)}}$ is the span of the color set $Z^{(i)}$, and the additional subspace $S_{Z^*}$ is the span of the sums of each edge in $Z$. To enlarge the $(\mathcal{A}, \mathcal{S})$-set, different matchings can be used, as long as they satisfy the following simple condition.

**Definition 29** Two matchings $\mathcal{X} = (X^{(0)}, \ldots, X^{(r-1)})$ and $\mathcal{Y} = (Y^{(0)}, \ldots, Y^{(r-1)})$ satisfy the pairing condition if any edge in $\mathcal{X}$ is monochromatic in $\mathcal{Y}$, and vice versa (see Figure 7.2).

![Figure 7.2](image)

Figure 7.2: Two matchings in the complete (2-uniform hyper)graph which satisfy the pairing condition. Each edge in one is monochromatic in the other, and vice versa.

Subspaces which correspond to distinct matchings $\mathcal{X} = (X^{(0)}, \ldots, X^{(r-1)})$ and $\mathcal{Y} = (Y^{(0)}, \ldots, Y^{(r-1)})$, that satisfy the pairing condition, have a useful property. This property is a corollary of the following lemma.
Lemma 70 If $D \in \{X^{(0)}, \ldots, X^{(r-1)}\}$ and $E \in \{Y^{(0)}, \ldots, Y^{(r-1)}\}$ then $|D \cap E| = \ell/r^2$.

Proof. Since $\mathcal{X}$ and $\mathcal{Y}$ satisfy the pairing condition, $D$ can be written as a union of edges from $\mathcal{Y}$, that is, $D = \bigcup_{i=0}^{\ell/r^2-1} \{y^{(0)}_i, y^{(1)}_i, \ldots, y^{(r-1)}_i\}$. Hence, $D$ contains exactly $\ell/r^2$ elements of $E$. ■

By [17, Lemma 11], in any MSR code which attains minimum repair bandwidth, the dimension of the intersection between any two repair subspaces is at most $\ell/r^2$. As a result of Lemma 70, we have that repair subspaces which correspond to different matchings attain this bound with equality.

Corollary 26 If $U \in \{S_{X^{(0)}}, S_{X^{(1)}}, \ldots, S_{X^{(r-1)}}, S_{X^*}\}$ and $V \in \{S_{Y^{(0)}}, S_{Y^{(1)}}, \ldots, S_{Y^{(r-1)}}, S_{Y^*}\}$, then $\dim(U \cap V) = \ell/r^2$.

Proof. We distinguish between the following cases.

Case 1. $U \in \{S_{X^{(0)}}, S_{X^{(1)}}, \ldots, S_{X^{(r-1)}}\}$ and $V \in \{S_{Y^{(0)}}, S_{Y^{(1)}}, \ldots, S_{Y^{(r-1)}}\}$. Let $B_U$ and $B_V$ be bases of $U$ and $V$, respectively, which consists of unit vectors only. Any $w \in U \cap V$ corresponds to a unique solution to the following equation, whose variables are $\{\alpha_i\}_{i=0}^{\ell/r^2-1}$ and $\{\beta_i\}_{i=0}^{\ell/r^2-1}$.

$$\sum_{u_i \in B_U} \alpha_i u_i = \sum_{v_i \in B_V} \beta_i v_i.$$

Since all vectors involved in this equation are unit vectors, and since by Lemma 70, we have that $|B_U \cap B_V| = \ell/r^2$, it follows that exactly $\ell/r^2$ of the coefficients in the left-hand side are equal to exactly $\ell/r^2$ in the right-hand side, and the rest of the coefficients are zero. Therefore, this equation has exactly $\ell/r^2$ degrees of freedom, and thus $\dim(U \cap V) = \ell/r^2$.

Case 2. $U \in \{S_{X^{(0)}}, S_{X^{(1)}}, \ldots, S_{X^{(r-1)}}\}$ and $V = S_{Y^*}$. Let $B_U$ be a basis of $U$ which consists of unit vectors only. As in the previous case, any
$w \in U \cap V$ corresponds to a solution of

$$\sum_{u_i \in B_U} \alpha_i u_i = \sum_{i=0}^{\ell/r-1} \beta_i (y_i^{(0)} + y_i^{(1)} + \ldots + y_i^{(r-1)}).$$

By Lemma 70, exactly $\ell/r^2$ of the edges $\{y_i^{(0)}, y_i^{(1)}, \ldots, y_i^{(r-1)}\}$ are in $B_U$, and hence, $(r-1)\ell/r^2$ of the coefficients in the right-hand side must be zero. The remaining $\ell/r^2$ coefficients may be chosen arbitrarily. Thus there are $\ell/r^2$ degrees of freedom for this equation, and hence $\dim(U \cap V) = \ell/r^2$.

**Case 3.** $U \in \{S_Y^{(0)}, S_Y^{(1)}, \ldots, S_Y^{(r-1)}\}$ and $V = S_X^*$. This case is symmetric to Case 2.

**Case 4.** $U = S_X^*$, $V = S_Y^*$. Any vector $w \in U \cap V$ corresponds to a solution to

$$\sum_{i=0}^{\ell/r-1} \alpha_i (x_i^{(0)} + x_i^{(1)} + \ldots + x_i^{(r-1)}) = \sum_{i=0}^{\ell/r-1} \beta_i (y_i^{(0)} + y_i^{(1)} + \ldots + y_i^{(r-1)}).$$

Any edge $\{x_i^{(0)}, x_i^{(1)}, \ldots, x_i^{(r-1)}\}$ is contained in either of $Y^{(0)}, Y^{(1)}, \ldots, Y^{(r-1)}$. Hence, for each value chosen for $\alpha_i$, exactly $r$ distinct values for the different $\beta$-s immediately follow. Hence, this equation also has $\ell/r^2$ degrees of freedom, and $\dim(U \cap V) = \ell/r^2$.

\[\blacksquare\]

**Remark 9** In the proof of Corollary 26, the subspace $S_X^*$ could equally be replaced with any subspace of the form

$$\left\langle \left\{ c_0 x_i^{(0)} + c_1 x_i^{(1)} + \cdots + c_{r-1} x_i^{(r-1)} \right\} \right\rangle_{i=0}^{\ell/r-1}$$

for any constants $c_i$ such that $c_i \neq 0$ for all $i$, and the subspace $S_Y^*$ could equally be replaced with any subspace of the form

$$\left\langle \left\{ d_0 y_i^{(0)} + d_1 y_i^{(1)} + \cdots + d_{r-1} y_i^{(r-1)} \right\} \right\rangle_{i=0}^{\ell/r-1}$$

for any constants $d_i$ such that $d_i \neq 0$ for all $i$. This is since cases 2-4 of the proof do not use the specific choice of $c_i = 1$ and $d_i = 1$. This fact will be used in the proof of Lemma 83 to follow.

171
In the sequel we use a large set of matchings in which every two matchings satisfy the pairing condition. To satisfy the nonsingular property (Definition 27), each matrix of the \((A, S)\)-set is multiplied by a properly chosen field constant, without compromising the invariance property and the independence property. The constructions of the \((A, S)\)-sets, which follow the general outline described in this subsection, are discussed in detail in the following sections.

7.B.4 Construction of an MSR Code with Two Parities

7.B.4.1 Two Parities Code from One Matching

Recall that the vertices of the complete graph \(K_\ell\) are identified by all unit vectors \(e_0, \ldots, e_{\ell-1}\) of length \(\ell, \ell = 2^m\) for some integer \(m\), and a matching \(Z = (Z, Z')\) is a set of \(\ell/2\) vertex-disjoint edges of \(K_\ell\). Such a matching will provide an \((A, S)\)-set of size 2, satisfying the subspace condition. The construction of this \((A, S)\)-set also relies on the following \(\ell \times \ell\) matrices, which resemble the matrix in (7.14). For \(\lambda \in \mathbb{F}_q^*\), consider the following two \(\ell/2 \times \ell/2\) matrices

\[
A^+(\lambda) \triangleq 
\begin{pmatrix}
0 & \lambda & 0 & 0 & \cdots & 0 & 0 \\
\lambda & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & 0 & \lambda & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & \lambda \\
0 & 0 & 0 & 0 & \cdots & 0 & \lambda \\
\end{pmatrix},
\]

and let \(A(\lambda)\) be the following \(\ell \times \ell\) block diagonal matrix

\[
A(\lambda) \triangleq 
\begin{pmatrix}
A^+(\lambda) & 0 \\
0 & A^{-}(\lambda) \\
\end{pmatrix}.
\]

The matrix \(A(\lambda)\) possesses several useful properties, which are essential in our construction. These useful properties follow from the fact that the minimal polynomial of \(A(\lambda)\) is \(x^2 - \lambda^2\). This form of the minimal polynomial
shows that the matrix $A(\lambda)$ acts as a transposition on the vectors of $\mathbb{F}_q^\ell$ which are not eigenvectors, up to a multiplication by $\lambda$. That is, all vectors which are not eigenvectors may be partitioned to pairs $(u, v)$ such that $u A(\lambda) = v$ and $v A(\lambda) = \lambda^2 u$, as proved in Lemma 71 which follows. In addition, for field with even characteristic, the matrix $A(\lambda)$ is non-diagonalizable. To the best of our knowledge, this constitutes the first construction of an $(A, S)$-set satisfying the subspace condition whose matrices are non-diagonalizable. Notice that the multiplication of a vector $v$ by the matrix $A(\lambda)$ switches between entries $2t$ and $2t+1$ of $v$ for all $t \in \{0, \ldots, \ell/2-1\}$, and multiplies all entries by either $\lambda$ or $-\lambda$ according to $t \leq \ell/4 - 1$ or $t > \ell/4 - 1$. This is demonstrated in the following lemma.

**Lemma 71** If $P \in \mathbb{F}_q^{\ell \times \ell}$ is an invertible matrix whose rows are $p_0, \ldots, p_{\ell-1}$, and $B \triangleq P^{-1} A(\lambda) P$ for some $\lambda \in \mathbb{F}_q^*$, then for all $t \in \{0, \ldots, \ell/2-1\}$

\[
p_{2t}B = \begin{cases} 
\lambda p_{2t+1} & \text{if } t \leq \ell/4 - 1 \\
-\lambda p_{2t+1} & \text{if } t > \ell/4 - 1 
\end{cases} \quad p_{2t+1}B = \begin{cases} 
\lambda p_{2t} & \text{if } t \leq \ell/4 - 1 \\
-\lambda p_{2t} & \text{if } t > \ell/4 - 1 
\end{cases}.
\]

Furthermore, the vectors $p_{2t+1} + p_{2t}$ and $p_{2t+1} - p_{2t}$ are eigenvectors of $B$.

**Proof.** By (7.16), for all $t \in \{0, \ldots, \ell/2 - 1\}$ we have that

\[
e_{2t}A(\lambda) = \begin{cases}
\lambda e_{2t+1} & \text{if } t \leq \ell/4 - 1 \\
-\lambda e_{2t+1} & \text{if } t > \ell/4 - 1 
\end{cases}, \quad e_{2t+1}A(\lambda) = \begin{cases}
\lambda e_{2t} & \text{if } t \leq \ell/4 - 1 \\
-\lambda e_{2t} & \text{if } t > \ell/4 - 1 
\end{cases}.
\]

In addition, since $PP^{-1} = I$, it follows that $p_{2t}P^{-1} = e_t$ for all $t \in \{0, \ldots, \ell/2 - 1\}$. Therefore, for all $t \in \{0, \ldots, \ell/2 - 1\}$

\[
p_{2t}B &= p_{2t}P^{-1}A(\lambda)P \\
&= e_{2t}A(\lambda)P \\
&= \pm \lambda e_{2t+1}P = \pm \lambda p_{2t+1}, \quad (7.17)
\]

\[
p_{2t+1}B &= p_{2t+1}P^{-1}A(\lambda)P \\
&= e_{2t+1}A(\lambda)P \\
&= \pm \lambda e_{2t}P = \pm \lambda p_{2t}, \quad (7.18)
\]

where the $\pm$ sign distinguishes between the cases $t \leq \ell/4 - 1$ and $t > \ell/4 - 1$. To see that $p_{2t+1} + p_{2t}$ and $p_{2t+1} - p_{2t}$ are eigenvectors of $B$, notice that by
adding and subtracting (7.17) and (7.18), we have that

\[(p_{2t+1} + p_{2t})B = \pm \lambda (p_{2t+1} + p_{2t}) \quad (7.19)\]
\[(p_{2t+1} - p_{2t})B = \mp \lambda (p_{2t+1} - p_{2t}). \quad (7.20)\]

Given a matching \(Z = (Z, Z')\), it is easily verified that the following two matrices are invertible, where \(z_i, z'_i\) are vertices in the complete graph, which are identified by unit vectors of length \(\ell\).

\[
P_Z \triangleq \begin{pmatrix}
    z_0 \\
    z'_0 - z_0 \\
    z_1 \\
    z'_1 - z_1 \\
    \vdots \\
    z_{\ell/2-1} \\
    z'_{\ell/2-1} - z_{\ell/2-1}
\end{pmatrix}, \quad P_{Z'} \triangleq \begin{pmatrix}
    z'_0 \\
    z_0 + z'_0 \\
    z'_1 \\
    z_1 + z'_1 \\
    \vdots \\
    z'_{\ell/2-1} \\
    z_{\ell/2-1} + z'_{\ell/2-1}
\end{pmatrix}
\]

**Definition 30** Given a matching \(Z = (Z, Z')\), let

\[
A_Z(\lambda) \triangleq P_Z^{-1} \cdot A(\lambda) \cdot P_Z, \quad S_Z \triangleq \langle Z \rangle = \{ \{ z_i \}_{i=0}^{\ell/2-1} \} \quad (7.21)
\]
\[
A_{Z'}(\lambda) \triangleq P_{Z'}^{-1} \cdot A(\lambda) \cdot P_{Z'}, \quad S_{Z'} \triangleq \langle Z' \rangle = \{ \{ z'_i \}_{i=0}^{\ell/2-1} \}. \quad (7.22)
\]

As an immediate consequence of Lemma 71 and Definition 30 we have the following.

**Corollary 27** For every \(i \in \{0, \ldots, \ell/4 - 1\}\),

\[
z_i A_Z(\lambda) = \begin{cases}
    \lambda (z'_i - z_i) & \text{if } i \leq \ell/4 - 1 \\
    -\lambda (z'_i - z_i) & \text{if } i > \ell/4 - 1
\end{cases},
\]
\[
z'_i A_{Z'}(\lambda) = \begin{cases}
    \lambda (z_i + z'_i) & \text{if } i \leq \ell/4 - 1 \\
    -\lambda (z_i + z'_i) & \text{if } i > \ell/4 - 1
\end{cases},
\]

and,

- For \(i \leq \ell/4 - 1\),
  - \(z'_i\) is an eigenvector of \(A_Z(\lambda)\) which corresponds to the eigenvalue \(\lambda\).
\( z_i \) is an eigenvector of \( A_{Z'}(\lambda) \) which corresponds to the eigenvalue \(-\lambda\).

- For \( i > \ell/4 - 1 \),

  - \( z'_i \) is an eigenvector of \( A_Z(\lambda) \) which corresponds to the eigenvalue \(-\lambda\).
  - \( z_i \) is an eigenvector of \( A_{Z'}(\lambda) \) which corresponds to the eigenvalue \( \lambda \).

A matching \( Z \) provides an \((A,S)\)-set of size two as follows.

**Lemma 72** If \( Z = (Z, Z') \) is a matching, then \{\( (A_Z(\lambda), S_Z), (A_{Z'}(\lambda), S_{Z'}) \)\} satisfies the subspace condition.

**Proof.** For convenience of notation, and since the proof which follows holds for each \( \lambda \neq 0 \), let \( A_Z \) and \( A_{Z'} \) denote \( A_Z(\lambda) \) and \( A_{Z'}(\lambda) \), respectively. We show that all properties of the subspace condition are satisfied.

To prove the independence property, notice that by Corollary 27,

\[
S_Z A_Z = \left\langle \{z'_i - z_i\}_{i=0}^{\ell/2-1} \right\rangle,
\]
and thus,

\[
S_Z A_Z + S_Z = S_{Z'} A_{Z'} + S_{Z'} = \mathbb{F}_q^\ell.
\]

To prove the invariance property, notice that by Corollary 27 \( S_Z \) (resp. \( S_{Z'} \)) is a span of eigenvectors\(^6\) of \( A_{Z'} \) (resp. \( A_Z \)) and hence it is \( A_{Z'} \) (resp. \( A_Z \)) invariant.

To prove the nonsingular property, first notice that \( A_Z, A_{Z'} \) are invertible since they are defined as a product of invertible matrices, and thus every \( 1 \times 1 \) block submatrix is invertible. Second, notice that

\[
\begin{pmatrix}
  I & I \\
  A_Z & A_{Z'}
\end{pmatrix}
\]
is invertible if and only if \( A_Z - A_{Z'} \) is invertible. Since \( Z \cup Z' \) is a basis of \( \mathbb{F}_q^{\ell} \), to show that \( A_Z - A_{Z'} \) is invertible it suffices to show that its image contains \( Z \cup Z' \).

\(^6\)Note that it does not comply with the definition of an eigenspace, since it contains vectors that correspond to distinct eigenvalues.
Let $i \in \{0, \ldots, \ell/2 - 1\}$, and notice that by Corollary 27 if $i \leq \ell/4 - 1$ then
\[
\lambda^{-1} z_i (A_Z - A_{Z'}) = \lambda^{-1} (z_i A_Z - z_i A_{Z'}) = \lambda^{-1} \left( \lambda (z_i' - z_i) + \lambda z_i \right) = z_i'
\]
\[
-\lambda^{-1} z_i' (A_Z - A_{Z'}) = -\lambda^{-1} \left( z_i' A_Z - z_i' A_{Z'} \right) = -\lambda^{-1} \left( \lambda z_i' - (z_i + z_i') \right) = z_i.
\]
On the other hand, if $i > \ell/4 + 1$, then
\[
-\lambda^{-1} z_i (A_Z - A_{Z'}) = -\lambda^{-1} (z_i A_Z - z_i A_{Z'}) = -\lambda^{-1} \left( -\lambda (z_i' - z_i) - \lambda z_i \right) = z_i'
\]
\[
\lambda^{-1} z_i' (A_Z - A_{Z'}) = \lambda^{-1} \left( z_i' A_Z - z_i' A_{Z'} \right) = \lambda^{-1} \left( -\lambda m_i' + \lambda (z_i' + z_i) \right) = z_i.
\]
Therefore, for all $i \in \{0, \ldots, \ell/2 - 1\}$, the vectors $z_i$ and $z_i'$ are in the image $\text{Im}(A_Z - A_{Z'})$, which implies that $A_Z - A_{Z'}$ is of full rank.

From Lemma 72 it is evident that any pair $(Z, \lambda)$ of a matching $Z = (Z, Z')$ and a nonzero field element $\lambda$ provides an $(A, S)$-set of size two. In Section 7.B.4.2 which follows we discuss the required relation between two such pairs $(X, \lambda_x)$, $(Y, \lambda_y)$ that allow the corresponding $(A, S)$-sets to be united without compromising the subspace condition.

### 7.B.4.2 Two Parities Code from Two Matchings

To construct larger $(A, S)$-sets, we analyse the required relations between two distinct pairs $(X, \lambda_x)$, $(Y, \lambda_y)$ of matchings $X = (X, X')$, $Y = (Y, Y')$ and field elements $\lambda_x$, $\lambda_y$, that allow the construction of an $(A, S)$-set of size four. In Lemma 73 which follows, we show that there exist three sufficient conditions that $(X, \lambda_x), (Y, \lambda_y)$ should satisfy for this purpose. The first condition states that $\lambda_x$ and $\lambda_y$ must be distinct. The second condition, called the pairing condition, appears in Definition 29. The third condition, which is a more subtle one and will only be relevant in fields with odd characteristic, is that the vertices of certain edges from $X$ fall into distinct halves defined by the order of $Y$, and vice versa.

Clearly, a set $\{ (X_i, \lambda_i) \}_{i=1}^t$ such that any two pairs satisfy all of the above
conditions, will provide an \((A,S)\)-set of size 2\(t\). In the sequel we provide such a set of size \(m\) over \(\mathbb{F}_q\), for any \(m \in \mathbb{N}\) and any \(q \geq m + 1\). This set will yield an \((A,S)\)-set of size 2\(m\) for \(q \geq m + 1\), which consists of matrices of size \(2^m \times 2^m\).

**Lemma 73** If \(\mathcal{X} = (X, X')\), \(\mathcal{Y} = (Y, Y')\) are matchings and \(\lambda_x, \lambda_y\) are nonzero field elements such that

1. \(\lambda_x \neq \lambda_y\).
2. \(\mathcal{X}\) and \(\mathcal{Y}\) satisfy the pairing condition (see Definition 29).
3. If \(\lambda_x = -\lambda_y\), then for all \(i \in \{0, \ldots, \ell/2 - 1\}\),
   - if \((x_i, x'_i) = (y_j, y_t)\) then \(i \leq \ell/4 - 1, j \leq \ell/4 - 1, \text{ and } t > \ell/4 - 1\), and
   - if \((x_i, x'_i) = (y'_j, y'_t)\) then \(i > \ell/4 - 1, j \leq \ell/4 - 1, \text{ and } t > \ell/4 - 1\).

then the \((A,S)\)-set

\[
\{(A_X(\lambda_x), S_X), (A_{X'}(\lambda_x), S_{X'}), (A_Y(\lambda_y), S_Y), (A_{Y'}(\lambda_y), S_{Y'})\}
\]

satisfies the subspace condition.

**Proof.** For convenience, we omit the notations of \(\lambda_x, \lambda_y\) from \(A_X(\lambda_x), A_{X'}(\lambda_x), A_Y(\lambda_y), A_{Y'}(\lambda_y)\) (even so \(\lambda_x\) and \(\lambda_y\) are crucial for this proof). The independence property follows directly from Lemma 72, as well as the non-singularity of any 1 \(\times\) 1 submatrix in the nonsingular property. To prove the invariance property, notice that the cases

\[
S_X A_{X'} = S_X \quad S_Y A_{Y'} = S_Y
\]

\[
S_{X'} A_X = S_{X'} \quad S_{Y'} A_Y = S_{Y'}
\]

follow from Lemma 72 as well. We prove now that \(S_X A_Y = S_X\), and the rest of the cases follow by symmetry.

Since \(S_X = \langle x_0, \ldots, x_{\ell/2 - 1} \rangle\), a necessary and sufficient condition for \(S_X A_Y = S_X\) is that \(x_i A_Y \in S_X\) for each \(i \in \{0, \ldots, \ell/2 - 1\}\). Let \(x_i \in S_X\) for some \(i \in \{0, \ldots, \ell/2 - 1\}\). Since \(\mathcal{X}\) and \(\mathcal{Y}\) are matchings over the same vertex set, we have that either \(x_i \in Y\) or \(x_i \in Y'\). If \(x_i \in Y\), i.e. \(x_i = y'_j\) for some \(j \in \{0, \ldots, \ell/2 - 1\}\), then by Corollary 27 and by the definition of \(A_Y\) (7.21), we have that \(y'_j\) is an eigenvector of \(A_Y\). Therefore,

\[
x_i A_Y = y'_j A_Y = \pm \lambda_y y'_j = \pm \lambda_y x_i \in S_X.
\]
On the other hand, if $x_i \in Y$, i.e. $x_i = y_j$ for some $j \in \{0, \ldots, \ell/2 - 1\}$, then by Corollary 27

$$x_i A_Y = y_j A_Y = \pm \lambda y (y'_j - y_j) = \pm \lambda y y'_j \mp \lambda y x_i. \quad (7.23)$$

According to B2 (the pairing condition), we have that if $y_j \in X$, then $y'_j \in X$ as well. Therefore (7.23) is a sum of two vectors in $S_X$, which implies that $x_i A_Y \in S_X$.

To prove the nonsingular property, we show that $X \cup X' \subseteq \text{Im} (A_X - A_Y)$, and the rest of the cases follow by symmetry. Since $X \cup X'$ is a basis of $F_{q^r}$, it will follow that $\text{rank}(A_X - A_Y) = \ell$ as required. We split the proof to two cases as follows.

**Case 1.** $\lambda_x \neq -\lambda_y$ (and thus $\lambda_x \neq \pm \lambda_y$ by B1). If $i \in \{0, \ldots, \ell/2 - 1\}$, then by A2, we have that either $(x_i, x'_i) = (y_j, y_t)$ or $(x_i, x'_i) = (y'_j, y'_t)$ for some distinct $j, t \in \{0, \ldots, \ell/2 - 1\}$. If $(x_i, x'_i) = (y'_j, y'_t)$, then simple calculations that follow from Corollary 27 show that

$$x_i(A_X - A_Y) = \begin{cases} \lambda x x'_i - (\lambda x + \lambda y) x_i & \text{if } i \leq \ell/4 - 1, j \leq \ell/4 - 1 \\ \lambda x x'_i - (\lambda x - \lambda y) x_i & \text{if } i \leq \ell/4 - 1, j > \ell/4 - 1 \\ -\lambda x x'_i + (\lambda x - \lambda y) x_i & \text{if } i > \ell/4 - 1, j \leq \ell/4 - 1 \\ -\lambda x x'_i + (\lambda x + \lambda y) x_i & \text{if } i > \ell/4 - 1, j > \ell/4 - 1 \end{cases} \quad (7.24)$$

$$x'_i(A_X - A_Y) = \begin{cases} (\lambda x - \lambda y) x'_i & \text{if } i \leq \ell/4 - 1, t \leq \ell/4 - 1 \\ (\lambda x + \lambda y) x'_i & \text{if } i \leq \ell/4 - 1, t > \ell/4 - 1 \\ -(\lambda x + \lambda y) x'_i & \text{if } i > \ell/4 - 1, t \leq \ell/4 - 1 \\ -(\lambda x - \lambda y) x'_i & \text{if } i > \ell/4 - 1, t > \ell/4 - 1 \end{cases} \quad (7.25)$$

Since $\lambda_x \neq \pm \lambda_y$, it follows by (7.25) that $x'_i \in \text{Im}(A_X - A_Y)$, which also implies by (7.24) that $x_i \in \text{Im}(A_X - A_Y)$. If $(x_i, x'_i) = (y_j, y_t)$,
then similar calculations show that

\[
x_i(A_X - A_Y) = \begin{cases} 
\lambda_x x'_i - (\lambda_x - \lambda_y)x_i - \lambda_y y'_j & \text{if } i \leq \ell/4 -1, j \leq \ell/4 -1 \\
\lambda_x x'_i - (\lambda_x + \lambda_y)x_i + \lambda_y y'_j & \text{if } i \leq \ell/4 -1, j > \ell/4 -1 \\
-\lambda_x x'_i + (\lambda_x + \lambda_y)x_i - \lambda_y y'_j & \text{if } i > \ell/4 -1, j \leq \ell/4 -1 \\
-\lambda_x x'_i + (\lambda_x - \lambda_y)x_i + \lambda_y y'_j & \text{if } i > \ell/4 -1, j > \ell/4 -1 
\end{cases}
\]

\[
x'_i(A_X - A_Y) = \begin{cases} 
(\lambda_x + \lambda_y)x'_i - \lambda_y y'_t & \text{if } i \leq \ell/4 -1, t \leq \ell/4 -1 \\
(\lambda_x - \lambda_y)x'_i + \lambda_y y'_t & \text{if } i \leq \ell/4 -1, t > \ell/4 -1 \\
-(\lambda_x - \lambda_y)x'_i - \lambda_y y'_t & \text{if } i > \ell/4 -1, t \leq \ell/4 -1 \\
-(\lambda_x + \lambda_y)x'_i + \lambda_y y'_t & \text{if } i > \ell/4 -1, t > \ell/4 -1 
\end{cases}.
\]

(7.26) (7.27)

Now, notice that since \( x_i = y_j \) we have that \( y_j \in X \). By B2, we also have that \( y'_j \in X \), and hence \( y'_j = x_s \) for some \( s \in \{0, \ldots, \ell/2 - 1\} \).

We have shown earlier that if \( x_s = y'_j \) then \( x_s = y'_j \in \text{Im}(A_X - A_Y) \).

Similarly, since \( x'_i = y_t \), we have that \( y_t \in X' \), i.e. \( y'_t = x'_r \) for some \( r \in \{0, \ldots, \ell/2 - 1\} \). This implies that \( x_r, x'_r \in Y' \) by the pairing condition, and thus, \( x'_r = y'_t \in \text{Im}(A_X - A_Y) \).

Since \( y'_t \in \text{Im}(A_X - A_Y) \), and since \( \lambda_x \neq \pm \lambda_y \), it follows from (7.27) that \( x'_i \in \text{Im}(A_X - A_Y) \). Therefore, by (7.26), and since \( y'_j \in \text{Im}(A_X - A_Y) \) and \( \lambda_x \neq \pm \lambda_y \), it follows that \( x_i \in \text{Im}(A_X - A_Y) \) as well.

**Case 2.** \( \lambda_x = -\lambda_y \) (and thus we have to consider B3). The pairing condition implies that either \( (x_i, x'_i) = (y_j, y_t) \) or \( (x_i, x'_i) = (y'_j, y'_t) \) for some distinct \( j, t \in \{0, \ldots, \ell/2 - 1\} \). However, by B3, most of the cases in (7.24), (7.25), (7.26), (7.27) are impossible. Hence, if \( (x_i, x'_i) = (y'_j, y'_t) \) then \( i > \ell/4 -1, j \leq \ell/4 -1, \) and \( t > \ell/4 -1, \) and thus

\[
x_i(A_X - A_Y) = -\lambda_x x'_i + (\lambda_x - \lambda_y)x_i
\]

\[
x'_i(A_X - A_Y) = (\lambda_y - \lambda_x)x'_i.
\]

Since \( \lambda_x \neq \lambda_y \), we have that \( x_i, x'_i \in \text{Im}(A_X - A_Y) \). If \( (x_i, x'_i) = (y_j, y_t) \) then \( i \leq \ell/4 -1, j \leq \ell/4 -1, \) and \( t > \ell/4 -1, \) and thus

\[
x_i(A_X - A_Y) = \lambda_x x'_i + (\lambda_y - \lambda_x)x_i - \lambda_y y'_j
\]

\[
x'_i(A_X - A_Y) = (\lambda_x - \lambda_y)x'_i + \lambda_y y'_t.
\]

As in Case 1, we can prove that \( y'_j, y'_t \in \text{Im}(A_X - A_Y) \), and get that since \( \lambda_x \neq \lambda_y \), we have that \( x_i, x'_i \in \text{Im}(A_X - A_Y) \).

179
By Lemma 7.3, we have that two matchings \( \mathcal{X}, \mathcal{Y} \) and two corresponding field elements \( \lambda_x, \lambda_y \) that meet the requirements B1-B3, provide an \((A,S)\)-set of size four. Therefore, a construction of a large set of pairs \((\mathcal{X}, \lambda_i)\), such that any two pairs satisfy B1-B3, is required for a construction of a large \((A,S)\)-set which satisfies the subspace condition.

### 7.B.4.3 Construction of Matchings for Two Parities

In the sequel, we construct a set \( \{(\mathcal{X}_i, \lambda_i)\}_{i=0}^{m-1} \) whose elements satisfy the requirements of Lemma 7.3 in pairs. We identify vertex \( e_i \) of \( K_\ell \) with the binary \( m \)-bit representation of \( i \). We will use the following standard notion of a boolean cube.

**Definition 31** Given a sequence of distinct indices \( i_1, \ldots, i_k \in \{0, \ldots, m-1\} \) and a sequence of binary values \( b_1, \ldots, b_k \in \{0, 1\} \), the boolean cube \( C(\{(i_j, b_j)\}_{j=1}^{k}) \), of size \( 2^{m-k} \), is the set of all \( m \)-bit vectors over \( \{0, 1\} \) that have \( b_j \) in entry \( i_j \) for all \( j = 1, \ldots, k \). That is,

\[
C(\{(i_j, b_j)\}_{j=1}^{k}) = \{x \in \{0, 1\}^m \mid \text{for all } j \in [k], \ x_{i_j} = b_j \}.
\]

We consider the elements in such a boolean cube as ordered according to the lexicographic order (see Example 8 below), that is, we consider a boolean cube as a sequence rather than a set.

**Example 8** If \( m = 4 \) then the boolean cube \( C(\{(1, 1), (2, 1)\}) \) is the set \( \{v_1, v_2, v_3, v_4\} \) such that \( (v_1, v_2, v_3, v_4) = (0110, 0111, 1110, 1111) \).

We begin by defining a set of matchings that meets the pairing condition.

**Definition 32** For any \( m \in \mathbb{N} \), define \( m \) matchings \( \{X_i, X'_i\}_{i=0}^{m-1} \) as follows

\[
X_{2t} : \left\{ \begin{array}{l}
X_{2t} = C(\{(2t, 0), (2t + 1, 0)\}) \circ C(\{(2t, 0), (2t + 1, 1)\}) \\
X'_{2t} = C(\{(2t, 1), (2t + 1, 0)\}) \circ C(\{(2t, 1), (2t + 1, 1)\})
\end{array} \right.
\]

\[
X_{2t+1} : \left\{ \begin{array}{l}
X_{2t+1} = C(\{(2t, 0), (2t + 1, 0)\}) \circ C(\{(2t, 1), (2t + 1, 0)\}) \\
X'_{2t+1} = C(\{(2t, 0), (2t + 1, 1)\}) \circ C(\{(2t, 1), (2t + 1, 1)\})
\end{array} \right.
\]
where \( t \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor - 1 \} \), and \( \circ \) indicates the concatenation of sequences.

If \( m \) is odd, we add the matching

\[
\begin{align*}
X_{m-1} &= C\left(\{(m-1, 0)\}\right) \\
X'_{m-1} &= C\left(\{(m-1, 1)\}\right)
\end{align*}
\]

**Example 9** If \( m = 4 \), then

\[
\begin{align*}
\mathcal{X}_0 : & \quad \begin{cases} 
X_0 = C(\{(0, 0), (1, 0)\}) \circ C(\{(0, 0), (1, 1)\}) \\
X'_0 = C(\{(0, 1), (1, 0)\}) \circ C(\{(0, 1), (1, 1)\})
\end{cases} \\
\mathcal{X}_1 : & \quad \begin{cases} 
X_1 = C(\{(0, 0), (1, 0)\}) \circ C(\{(0, 1), (1, 0)\}) \\
X'_1 = C(\{(0, 0), (1, 1)\}) \circ C(\{(0, 1), (1, 1)\})
\end{cases} \\
\mathcal{X}_2 : & \quad \begin{cases} 
X_2 = C(\{(2, 0), (3, 0)\}) \circ C(\{(2, 1), (3, 1)\}) \\
X'_2 = C(\{(2, 1), (3, 0)\}) \circ C(\{(2, 1), (3, 1)\})
\end{cases} \\
\mathcal{X}_3 : & \quad \begin{cases} 
X_3 = C(\{(2, 0), (3, 0)\}) \circ C(\{(2, 1), (3, 0)\}) \\
X'_3 = C(\{(2, 0), (3, 1)\}) \circ C(\{(2, 1), (3, 1)\})
\end{cases}
\]

which implies that

\[
\begin{align*}
\mathcal{X}_0 : & \quad \begin{cases} 
X_0 = (0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111) \\
X'_0 = (1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111)
\end{cases} \\
\mathcal{X}_1 : & \quad \begin{cases} 
X_1 = (0000, 0001, 0010, 0011, 1000, 1001, 1010, 1011) \\
X'_1 = (0100, 0101, 0110, 0111, 1100, 1101, 1110, 1111)
\end{cases} \\
\mathcal{X}_2 : & \quad \begin{cases} 
X_2 = (0000, 0100, 1000, 1100, 0001, 0101, 1001, 1101) \\
X'_2 = (0010, 0110, 1010, 1110, 0011, 0111, 1011, 1111)
\end{cases} \\
\mathcal{X}_3 : & \quad \begin{cases} 
X_3 = (0000, 0100, 1000, 1100, 0010, 0110, 1010, 1110) \\
X'_3 = (0001, 0101, 1001, 1101, 0011, 0111, 1011, 1111)
\end{cases}
\]

where the values in bold indicate the fixed entries in each boolean cube.

Before we choose a field element for each matching, which satisfies B1-B3 of Lemma \[73\] we prove that the matchings from Definition \[32\] satisfy the pairing condition.

**Lemma 74** Each two distinct matchings \( \mathcal{X}_i, \mathcal{X}_j \) from Definition \[32\] satisfy the pairing condition.
Proof. Denote the elements of the matchings $X_i, X_j$ as

$$X_i = (x_{i,0}, \ldots, x_{i,\ell/2-1})$$

$$X'_i = (x'_{i,0}, \ldots, x'_{i,\ell/2-1})$$

$$X_j = (x_{j,0}, \ldots, x_{j,\ell/2-1})$$

$$X'_j = (x'_{j,0}, \ldots, x'_{j,\ell/2-1}).$$

By Definition 32, it is evident that in every edge $(x_{i,t}, x'_{i,t}) \in X_i$, the $i$-th entry of $x_{i,t}$ is 0, the $i$-th entry of $x'_{i,t}$ is 1, and the rest of the entries are identical. Similarly, in every edge $(x_{j,t}, x'_{j,t}) \in X_j$, the $j$-th entry of $x_{j,t}$ is 0, the $j$-th entry of $x'_{j,t}$ is 1, and the rest of the entries are identical. Therefore, for every edge $(x_{i,t}, x'_{i,t}) \in X_i$, if the $j$-th entry of both $x_{i,t}$ and $x'_{i,t}$ is 0, then $x_{i,t}, x'_{i,t} \in X_j$, and if it is 1, then $x_{i,t}, x'_{i,t} \in X'_j$. Therefore, $X_j$ is a union of edges from $X_i$. The proof that $X_i$ is a union of edges from $X'_j$ is similar.

We now turn to choose a proper nonzero field element for every matching from Definition 32. This choice must comply with requirements B1 and B3 of Lemma 73. Note that if $q$ is even, then B3 follows from B1 (vacuously). Hence, if the field characteristics is 2, the choice of field elements is straightforward.

Lemma 75 If $q \geq m + 1$ is a power of two, then by any arbitrary choice of pairwise distinct elements from $\mathbb{F}_q^*$ for the $m$ matchings from Definition 32, the resulting $(A, S)$-set satisfies B1-B3 from Lemma 73.

Proof. Since the assigned elements are distinct, every two matchings satisfy property B1 of Lemma 73. According to Lemma 74, every two matchings satisfy the pairing condition (B2) as well. Since $q$ is even, property B3 is implied by property B1.

If $q$ is odd, more care is needed for the mapping of nonzero field elements to the matchings. We do this by choosing field elements $\lambda$ and $-\lambda$ for two adjacent matchings $X_{2t}, X_{2t+1}$.

Lemma 76 Let $q \geq m+1$ be a power of an odd prime. Assume we are given an arbitrary choice of pairwise distinct elements from $\mathbb{F}_q^*$ to the $m$ matchings from Definition 32, such that for $X_{2t}, X_{2t+1}$, some elements $\lambda, -\lambda$ are chosen for every $t \in \{0, \ldots, \lfloor m/2 \rfloor - 1\}$. Then, the resulting $(A, S)$-set satisfies B1-B3 from Lemma 73.
Proof. For every two distinct matchings, requirement B1 of Lemma 73 is trivially satisfied, and requirement B2 is satisfied by Lemma 74. To prove B3, let $\lambda_i = -\lambda_j$ be two field elements which are chosen for two matchings $\mathcal{X}_i, \mathcal{X}_j$. Without loss of generality (w.l.o.g.) assume that $i = 2t$ and $j = 2t+1$ for some $t \in \{0, \ldots, \lfloor \frac{n}{2} \rfloor - 1\}$.

Let $(x'_{2t,s}, x'_{2t,s})$ be an edge in $\mathcal{X}_{2t}$, which implies that the $(2t)$-th bit of $x_{2t,s}$ is 0 and the $(2t)$-th bit of $x'_{2t,s}$ is 1. To prove B3, we must show that

if $(x_{2t,s}, x'_{2t,s}) = (x_{2t+1,u}, x_{2t+1,r})$ then $s \leq \ell/4 - 1$, $u \leq \ell/4 - 1$, and $r > \ell/4 - 1$, and

if $(x_{2t,s}, x'_{2t,s}) = (x'_{2t+1,u}, x'_{2t+1,r})$ then $s > \ell/4 - 1$, $u \leq \ell/4 - 1$, and $r > \ell/4 - 1$.

If $(x_{2t,s}, x'_{2t,s}) = (x_{2t+1,u}, x_{2t+1,r})$ for some $u, r \in \{0, \ldots, \ell/2 - 1\}$, it follows that the $(2t)$-th bit of $x_{2t+1,s}$ and $x'_{2t+1,s}$ is 0. Therefore

$$x_{2t,s} = x_{2t+1,u} \in C(\{(2t,0), (2t+1,0)\})$$

$$x'_{2t,s} = x_{2t+1,r} \in C(\{(2t,1), (2t+1,0)\}),$$

and hence, by the definition of $\mathcal{X}_{2t+1}$ (Definition 32), it follows that $u \leq \ell/4 - 1$ and $r > \ell/4 - 1$. In addition, by the definition of $\mathcal{X}_{2t}$ it follows that $s \leq \ell/4 - 1$.

If $(x_{2t,s}, x'_{2t,s}) = (x'_{2t+1,u}, x'_{2t+1,r})$ for some $u, r \in \{0, \ldots, \ell/2 - 1\}$, it follows that the $(2t)$-th bit of $x_{2t+1,s}$ and $x'_{2t+1,s}$ is 1. Therefore

$$x_{2t,s} = x'_{2t+1,u} \in C(\{(2t,0), (2t+1,1)\})$$

$$x'_{2t,s} = x'_{2t+1,r} \in C(\{(2t,1), (2t+1,1)\}),$$

and hence, by the definition of $\mathcal{X}_{2t+1}$, it follows that $u \leq \ell/4 - 1$ and $r > \ell/4 - 1$. In addition, by the definition of $\mathcal{X}_{2t}$ it follows that $s > \ell/4 - 1$.  

The main construction of this section is summarized in the following theorem.

**Theorem 38** If $m$ is a positive integer and $q \geq m + 1$ is a prime power, then there exists an explicitly defined $(A,S)$-set $C$ of size $2m$ and $2^m \times 2^m$ matrices over $\mathbb{F}_q$, which satisfies the subspace condition.

**Proof.** Let $\{\mathcal{X}_i = (X_i, X'_i)\}_{i=0}^{m-1}$ be the set of matchings from Definition 32, which by Lemma 74 satisfies the pairing condition (Definition 29).
If $q$ is even, then let $\lambda_0, \ldots, \lambda_{m-1}$ be distinct elements in $\mathbb{F}_q^*$, and let

$$
\mathcal{C} \triangleq \bigcup_{i=0}^{m-1} \left\{ (A_X(\lambda_i), S_X), (A_X'(\lambda_i), S_X') \right\},
$$

(7.28)

where $(A_X(\lambda_i), S_X), (A_X'(\lambda_i), S_X')$ were defined in Lemma 72. Since conditions B1-B3 of Lemma 73 are met with respect to every two matchings and their respective field elements, it follows that $\mathcal{C}$ satisfies the subspace condition.

If $q$ is odd, let $\lambda_0, \ldots, \lambda_{m-1}$ be distinct elements in $\mathbb{F}_q^*$ such that $\lambda_{2t} = -\lambda_{2t+1}$ for every $t \in \{0, \ldots, \lfloor \frac{m}{2} \rfloor - 1\}$. Define $\mathcal{C}$ in a similar way to the one defined in (7.28). Conditions B1 and B2 are satisfied as in the case of an even $q$. Condition B3 is satisfied by Lemma 76, and therefore $\mathcal{C}$ satisfies the subspace condition in this case as well.

Notice that since each repair subspace in Theorem 38 contains a basis of unit vectors, it follows that the resulting code has the access-optimal property (see Section 7.B.2.2). Moreover, it is readily verified that the resulting code attains the sub-packetization bound for access-optimal codes (7.8).

### 7.B.5 Construction of an MSR Code with Three Parities

In this section we construct MSR codes with three parities using the framework mentioned in Subsection 7.B.3. The size of the matrices is $\ell \times \ell$, where $\ell = 3^m$ for some integer $m$. This construction requires that all three roots of unity of order three lie in the base field (which implies the necessary condition $3|q - 1$). If $q$ is odd we require that $q \geq 6m + 1$ and if $q$ is even we require that $q \geq 3m + 1$. As the roots of unity of order three play an important role in this section, recall the following properties of these roots, some of which can be generalized for every set of roots of unity of any order.

#### Lemma 77

If $q$ is a prime power such that $3|q - 1$, then $\mathbb{F}_q$ contains three distinct roots of unity of order three $1, \gamma_1, \gamma_2$, which satisfy $1 + \gamma_1 + \gamma_2 = 0$, $\gamma_1^2 = \gamma_2$, and $\gamma_2^{-1} = \gamma_1$.

**Proof.** The existence of all roots of unity in $\mathbb{F}_q$ is a consequence of the Sylow Theorems [3 Section XII.5]. In addition, it is widely known that the
sum of all roots of unity of any order is 0 [8, Chapter 2, Ex. 2.49]. The other properties follow from the fact that \( \{1, \gamma_1, \gamma_2\} \) is a multiplicative subgroup of \( \mathbb{F}_q^* \).

From now on we assume that \( 3 | q - 1 \), and \( 1, \gamma_1, \gamma_2 \) are the three roots of unity of order three. Notice that this necessary condition rules out the possibility of using fields with characteristic 3.

Proving the nonsingular property for three parities becomes more involved, since we must show that any \( 1 \times 1 \), \( 2 \times 2 \), and \( 3 \times 3 \) block submatrix of (7.13) is invertible. Fortunately, showing that any \( 1 \times 1 \) block submatrix (that is, an entry in (7.13)) is invertible is trivial in our construction. Moreover, assuming that any entry of (7.13) is invertible, and that submatrices of the form

\[
\begin{pmatrix}
I & A_i \\
I & A_j
\end{pmatrix}
\]

are invertible, by using block-row operations\(^7\) we have that matrices of the form

\[
\begin{pmatrix}
A_i & A_i^2 \\
A_j & A_j^2
\end{pmatrix}
\]

are invertible as well. That is, by multiplying the top row of the latter matrix by \( A_i^{-1} \) and multiplying the bottom row by \( A_j^{-1} \), we get the former matrix. Hence, for three parities, to show that an \((\mathcal{A}, \mathcal{S})\)-set satisfies the nonsingular it suffices to show the following three conditions.

**Conditions for the nonsingular property:** (three parities)

1. For all \( i, j \in [k], i \neq j \), the matrix \( \begin{pmatrix} I & A_i \\ I & A_j \end{pmatrix} \) is invertible.

2. For all \( i, j \in [k], i \neq j \), the matrix \( \begin{pmatrix} I & A_i^2 \\ I & A_j^2 \end{pmatrix} \) is invertible.

\(^7\)The three standard block operations are interchanging two block rows (columns), multiplying a block row (column) from the left (right) by a non-singular matrix, and multiplying a block row (column) by a matrix from the left (right) and adding it to another row.
3. For all distinct \(i, j, t \in [k]\), the matrix
\[
\begin{pmatrix}
I & A_i & A_i^2 \\
I & A_j & A_j^2 \\
I & A_t & A_t^2
\end{pmatrix}
\]
is invertible.

### 7.B.5.1 Three Parities from One Matching

Recall that in Section 7.B.4, every matching \(Z\) (Definition 28) provided an \((A, S)\)-set \((A_Z, S_Z), (A_{Z'}, S_{Z'})\), where \(S_Z\) is an eigenspace of \(A_{Z'}\) and \(S_{Z'}\) is an eigenspace of \(A_Z\). Later on, we added together \((A, S)\)-sets which were defined by different matchings satisfying the pairing condition (Definition 29).

For three parities, we consider the natural generalization of matchings in the complete 3-uniform hypergraph.

Similarly to Section 7.B.4, this construction will rely on \(\ell \times \ell\) matrices whose minimal polynomial is \(x^3 - \lambda^3\) for some \(\lambda \in \mathbb{F}_q^*\). All the matrices in the \((A, S)\)-set will be similar to the matrix \(A\) (7.14), which for \(r = 3\) takes the form of

\[
A = \begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0
\end{pmatrix}.
\] (7.29)

According to Corollary 25 we have the following lemma.

**Lemma 78** The matrix \(A\) (7.29) is diagonalizable, with the following eigenspaces,
1. For the eigenvalue 1, a basis of the eigenspace is
\[
\{ \begin{array}{c}
(1, 1, 1, 0, 0, \ldots, 0, 0, 0), \\
(0, 0, 0, 1, 1, \ldots, 0, 0, 0), \\
\ldots \\
(0, 0, 0, 0, 0, \ldots, 1, 1, 1)
\end{array} \}.
\]

2. For the eigenvalue \( \gamma_1 \), a basis of the eigenspace is
\[
\{ \begin{array}{c}
(1, \gamma_1, \gamma_2, 0, 0, \ldots, 0, 0, 0), \\
(0, 0, 0, 1, \gamma_1, \gamma_2, \ldots, 0, 0, 0), \\
\ldots \\
(0, 0, 0, 0, 0, \ldots, 1, \gamma_1, \gamma_2)
\end{array} \}.
\]

3. For the eigenvalue \( \gamma_2 \), a basis of the eigenspace is
\[
\{ \begin{array}{c}
(1, \gamma_2, \gamma_1, 0, 0, \ldots, 0, 0, 0), \\
(0, 0, 0, 1, \gamma_2, \gamma_1, \ldots, 0, 0, 0), \\
\ldots \\
(0, 0, 0, 0, 0, \ldots, 1, \gamma_2, \gamma_1)
\end{array} \}.
\]

In addition, the subspace \( S \triangleq \langle e_0, e_3, e_6, \ldots \rangle \) is an independent subspace of \( A \).

The matrices in our \((A, S)\)-set are similar to a constant multiple of the matrix \( A \), and thus they are also diagonalizable. The structure of their eigenspaces, which follows from Lemma 78, is as follows.

**Lemma 79** If \( P \in F_q^{\ell \times \ell} \) is an invertible matrix whose rows are \( p_0, \ldots, p_{\ell-1} \), and \( M \equiv \lambda P^{-1} A P \) for some \( \lambda \in F_q^* \), then \( M \) has the following eigenspaces,

1. For the eigenvalue \( \lambda \), a basis of the eigenspace is \( \{ p_{\ell i} + p_{\ell i+1} + p_{\ell i+2} \mid i \in \{0, \ldots, \ell/3 - 1\} \} \).

2. For the eigenvalue \( \gamma_1 \lambda \), a basis of the eigenspace is \( \{ p_{\ell i} + \gamma_1 p_{\ell i+1} + \gamma_2 p_{\ell i+2} \mid i \in \{0, \ldots, \ell/3 - 1\} \} \).
3. For the eigenvalue $\gamma_2\lambda$, a basis of the eigenspace is \( \{ p_{3i} + \gamma_2 p_{3i+1} + \gamma_1 p_{3i+2} \mid i \in \{0, \ldots, \ell/3 - 1\} \} \).

In addition, the subspace \( S \triangleq \langle p_0, p_3, p_6, \ldots \rangle \) is an independent subspace of \( M \).

We are now in a position to describe the \((A, S)\)-set, of size three, that is given by a single-matching. \((A, S)\)-sets that are given by a union of single-matching \((A, S)\)-sets will be discussed in the sequel. As mentioned earlier, all three matrices of this \((A, S)\)-set are similar to \( A \). The \( \ell \times \ell \) change matrices are defined using \( 3 \times \ell \) constituent blocks (see (7.15)) as follows. For \( \alpha, \beta \in \mathbb{F}_q^* \) and \( u, v, w \in \mathbb{F}_q^\ell \), let

\[
N(\alpha, \beta, u, v, w) = \begin{pmatrix}
1 & 0 & 0 \\
1 & -\frac{\alpha \gamma_1}{\gamma_1 - 1} & \frac{\beta}{\gamma_1 - 1} \\
1 & \frac{\alpha}{\gamma_1 - 1} & -\frac{\beta \gamma_1}{\gamma_1 - 1}
\end{pmatrix}
\cdot
\begin{pmatrix}
u \\
w
\end{pmatrix}.
\]

(7.30)

The determinant of the \( 3 \times 3 \) matrix in (7.30) equals \( \alpha \beta \cdot \frac{\gamma_1 + 1}{\gamma_1 - 1} \), which is nonzero, and thus \( N(\alpha, \beta, u, v, w) \) is row-equivalent to a matrix whose rows are \( u, v, w \), for any choice of \( \alpha, \beta \in \mathbb{F}_q^* \). This fact gives rise to the following necessary lemma, which can be easily proved.

**Lemma 80** If \( Z = (Z, Z', Z'') \) is a matching, then for any choice of \( \alpha, \alpha', \alpha'' \)
and \( \beta, \beta', \beta'' \), in \( \mathbb{F}_q^* \), the following matrices are invertible.

\[
P_Z \triangleq \begin{pmatrix}
N(\alpha, \beta, z_0, z_0') \\
N(\alpha, \beta, z_1, z_1') \\
\vdots \\
N(\alpha, \beta, z_{\ell/3-1}, z_{\ell/3-1}')
\end{pmatrix},
\]

\[
P_{Z'} \triangleq \begin{pmatrix}
N(\alpha', \beta', z_0', z_0) \\
N(\alpha', \beta', z_1', z_1) \\
\vdots \\
N(\alpha', \beta', z_{\ell/3-1}', z_{\ell/3-1})
\end{pmatrix},
\]

\[
P_{Z''} \triangleq \begin{pmatrix}
N(\alpha'', \beta'', z_0'', z_0') \\
N(\alpha'', \beta'', z_1'', z_1') \\
\vdots \\
N(\alpha'', \beta'', z_{\ell/3-1}'', z_{\ell/3-1}')
\end{pmatrix}
\]

**Lemma 81** If \( Z = (Z, Z', Z'') \) is a matching, then for any \( \lambda \in \mathbb{F}_q^* \), the following \((A,S)\)-set satisfies the subspace condition.

\[
A_Z(\lambda) \triangleq \lambda \cdot P_Z^{-1}AP_Z, \quad S_Z \triangleq \langle Z \rangle
\]

\[
A_{Z'}(\lambda) \triangleq \lambda \cdot P_{Z'}^{-1}AP_{Z'}, \quad S_{Z'} \triangleq \langle Z' \rangle
\]

\[
A_{Z''}(\lambda) \triangleq \lambda \cdot P_{Z''}^{-1}AP_{Z''}, \quad S_{Z''} \triangleq \langle Z'' \rangle
\]

where \( \alpha, \alpha', \alpha'', \beta, \beta', \beta'' \) are nonzero field elements that will be chosen according to the field characteristic.

**Proof.** For convenience of notation, denote \( A_Z(\lambda), A_{Z'}(\lambda), \) and \( A_{Z''}(\lambda) \) by \( A_Z, A_{Z'}, \) and \( A_{Z''}, \) respectively. To prove the independence property, it follows from Lemma 79 that for any matrix of the form \( P^{-1}AP \), where the rows of \( P \) are \( \{p_0, \ldots, p_{\ell-1}\} \), the subspace \( \langle p_0, p_3, \ldots \rangle \) is an independent subspace of \( P^{-1}AP \). Notice that the vectors in \( P_Z \) that correspond to rows \( p_i \) with \( i \equiv 0 \mod 3 \) are \( \{z_0, \ldots, z_{\ell/3-1}\} \). Hence, \( S_Z \) is an independent subspace of \( A_Z \). Similarly, we have that \( S_{Z'}, S_{Z''} \) are independent subspaces.
of $A_{Z'}, A_{Z''}$, respectively. Therefore, the independence property is satisfied.

To show the invariance property, the eigenspaces of $A_Z, A_{Z'},$ and $A_{Z''}$, are computed according to Lemma 79. The eigenspace of $A_Z$ that corresponds to the eigenvalue $\gamma_1 \lambda$ is the span of vectors of the form $p_{3i} + \gamma_1 p_{3i+1} + \gamma_2 p_{3i+2}$, for $i \in \{0, \ldots, \ell/3 - 1\}$, where $p_j$ is the $j$-th row of $P_Z$. Therefore, by the definition of $N(7.30)$, we have that the eigenspace of $A_Z$ that corresponds to the eigenvalue $\gamma_1 \lambda$ is the span of

$$\left\{ z_i + \gamma_1 \cdot \left( z_i - \frac{\alpha \gamma_1}{\gamma_1 - 1} z_i' + \frac{\beta}{\gamma_1 - 1} z_i'' \right) \right\}$$

and clearly, this is $S_{Z'}$. Similarly, we have that the eigenspace of $A_Z$ which corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_{Z''}$. Hence, the subspaces $S_{Z'}$ and $S_{Z''}$ are invariant subspaces of $A_Z$. By identical arguments, it can be shown that for $A_{Z'}$, the eigenspace that corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_{Z''}$ and the eigenspace that corresponds to the eigenvalue $\gamma_1 \lambda$ is $S_{Z'}$. Furthermore, the eigenspace of $A_{Z''}$ that corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_Z$, and the eigenspace that corresponds to the eigenvalue $\gamma_1 \lambda$ is $S_{Z'}$. Therefore, the invariance property holds.

To show the nonsingular property, we show that Conditions 113 are met. To prove Condition 1, it should be shown that $\text{rank}(A_Z - A_{Z''}) = \ell$, $\text{rank}(A_Z - A_{Z'}) = \ell$, and $\text{rank}(A_{Z'} - A_{Z''}) = \ell$. To show that $\text{rank}(A_Z - A_{Z''}) = \ell$, it will be proved that for all $0 \leq i \leq \ell/3 - 1$, $\{z_i, z_i', z_i''\} \subseteq \text{Im}(A_Z - A_{Z''})$. Since $Z \cup Z' \cup Z''$ is a basis of $\mathbb{F}_q^\ell$, the claim will follow.

By the structure of the matrix $A$, it can be easily verified that if $P$ is an invertible matrix whose rows are $\{p_0, \ldots, p_{\ell-1}\}$, then

$$p_{3i} \lambda P^{-1} A P = \lambda p_{3i+2}$$
$$p_{3i+1} \lambda P^{-1} A P = \lambda p_{3i}$$
$$p_{3i+2} \lambda P^{-1} A P = \lambda p_{3i+1},$$

and clearly, this is $S_{Z'}$. Similarly, we have that the eigenspace of $A_Z$ which corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_{Z''}$. Hence, the subspaces $S_{Z'}$ and $S_{Z''}$ are invariant subspaces of $A_Z$. By identical arguments, it can be shown that for $A_{Z'}$, the eigenspace that corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_{Z''}$ and the eigenspace that corresponds to the eigenvalue $\gamma_1 \lambda$ is $S_{Z'}$. Furthermore, the eigenspace of $A_{Z''}$ that corresponds to the eigenvalue $\gamma_2 \lambda$ is $S_Z$, and the eigenspace that corresponds to the eigenvalue $\gamma_1 \lambda$ is $S_{Z'}$. Therefore, the invariance property holds.

To show the nonsingular property, we show that Conditions 113 are met. To prove Condition 1, it should be shown that $\text{rank}(A_Z - A_{Z''}) = \ell$, $\text{rank}(A_Z - A_{Z'}) = \ell$, and $\text{rank}(A_{Z'} - A_{Z''}) = \ell$. To show that $\text{rank}(A_Z - A_{Z''}) = \ell$, it will be proved that for all $0 \leq i \leq \ell/3 - 1$, $\{z_i, z_i', z_i''\} \subseteq \text{Im}(A_Z - A_{Z''})$. Since $Z \cup Z' \cup Z''$ is a basis of $\mathbb{F}_q^\ell$, the claim will follow.

By the structure of the matrix $A$, it can be easily verified that if $P$ is an invertible matrix whose rows are $\{p_0, \ldots, p_{\ell-1}\}$, then
and therefore,

\[ z_i A_Z = \lambda \left( z_i + \frac{\alpha}{\gamma_1 - 1} z_i' - \frac{\beta \gamma_1}{\gamma_1 - 1} z_i'' \right) \]

\[ z_i'' A_{Z''} = \lambda \left( z_i'' + \frac{\alpha''}{\gamma_1 - 1} z_i - \frac{\beta'' \gamma_1}{\gamma_1 - 1} z_i'' \right). \]

Hence, since \( S_Z, S_{Z''} \) are eigenspaces of \( A_Z \) and \( S_{Z'}, S_{Z''} \) are eigenspaces of \( A_{Z''} \), it follows that

\[ z_i (A_Z - A_{Z''}) = z_i A_Z - z_i A_{Z''} \]

\[ = \lambda \left( z_i + \frac{\alpha}{\gamma_1 - 1} z_i' - \frac{\beta \gamma_1}{\gamma_1 - 1} z_i'' \right) - \gamma_2 \lambda z_i \]

\[ = \lambda \left( (1 - \gamma_2) z_i + \frac{\alpha}{\gamma_1 - 1} z_i' - \frac{\beta \gamma_1}{\gamma_1 - 1} z_i'' \right) \]

\[ z_i' (A_Z - A_{Z''}) = z_i' A_Z - z_i' A_{Z''} = \lambda (\gamma_2 - \gamma_1) z_i' \]

\[ z_i'' (A_Z - A_{Z''}) = z_i'' A_Z - z_i'' A_{Z''} \]

\[ = \gamma_1 \lambda z_i'' - \lambda \left( z_i'' + \frac{\alpha''}{\gamma_1 - 1} z_i - \frac{\beta'' \gamma_1}{\gamma_1 - 1} z_i'' \right) \]

\[ = \lambda \left( -\frac{\alpha''}{\gamma_1 - 1} z_i + \frac{\beta'' \gamma_1}{\gamma_1 - 1} z_i' + (\gamma_1 - 1) z_i'' \right) \]

which in a more convenient matrix notation becomes

\[
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
\cdot (A_Z - A_{Z''}) = \lambda \begin{pmatrix}
  1 - \gamma_2 & \alpha & -\beta \gamma_1 \\
  0 & \gamma_2 - \gamma_1 & 0 \\
  -\frac{\alpha''}{\gamma_1 - 1} & \frac{\beta'' \gamma_1}{\gamma_1 - 1} & \gamma_1 - 1
\end{pmatrix}
\cdot
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix} \triangleq \lambda \cdot \Phi \cdot
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}.
\]

(7.31)

To show that \( z_i, z_i', z_i'' \in \text{Im}(A_Z - A_{Z''}) \), it suffices to show that \( \Phi \) is invertible. Since \( \lambda (\gamma_2 - \gamma_1) \neq 0 \), it is enough to prove that

\[
\begin{pmatrix}
  1 - \gamma_2 & -\beta \gamma_1 \\
  -\frac{\alpha''}{\gamma_1 - 1} & \gamma_1 - 1
\end{pmatrix}
\]

(7.32)

is invertible. Simple calculations, which follow from the properties of \( \gamma_1, \gamma_2 \) (Lemma 77) show that this matrix has a nonzero determinant if and only if

\[ * \text{Any } n \in \mathbb{N} \text{ denotes the summation of } n \text{ copies of the unity of the field } \mathbb{F}_q. \]
α"β \neq 9. Similar arguments show that \( \text{rank}(A_Z - A_{Z''}) = \ell \) if and only if \( \alpha'\beta'' \neq 9 \) and \( \alpha\beta \neq 9 \), respectively. Proper \( \alpha, \alpha', \alpha'', \beta, \beta', \beta'' \) will be chosen is the sequel.

To show Condition 2, it must proved that the difference between any two squares of matrices in the \((A, S)\)-set has full rank. Fortunately, the squares of the matrices in the \((A, S)\)-set present a very similar behavior to the matrices themselves. That is, if we denote any matrix in the \((A, S)\)-set by \( \hat{A} = \lambda P^{-1} \), then \( \hat{A}^2 = \lambda^2 P^{-1} A^2 P \), and

\[
\text{if } v\hat{A} = \lambda_1 v, \text{ then } v\hat{A}^2 = \lambda_2 v, \\
\text{if } v\hat{A}_i = \lambda_2 v, \text{ then } v\hat{A}^2 = \lambda_1 v.
\]

Hence, if \( S_1, S_2 \) are the eigenspaces of \( \hat{A} \) that correspond to the eigenvalues \( \lambda_1, \lambda_2 \), respectively, then the eigenspaces of \( \hat{A}^2 \) that correspond to the eigenvalues \( \lambda_1^2, \lambda_2^2 \) are \( S_2, S_1 \), respectively. Moreover, we have that

\[
\begin{align*}
z_iA_Z^2 &= \lambda^2 \left( z_i - \frac{\alpha\gamma_1}{\gamma_1 - 1} z'_i + \frac{\beta}{\gamma_1 - 1} z''_i \right), \\
z''_iA_{Z''}^2 &= \lambda^2 \left( z''_i - \frac{\alpha''\gamma_1}{\gamma_1 - 1} z_i + \frac{\beta''}{\gamma_1 - 1} z''_i \right),
\end{align*}
\]

and hence,

\[
\begin{align*}
z_i(A_Z^2 - A_{Z''}^2) &= \lambda^2 \left( (1 - \gamma_1) z_i - \frac{\alpha\gamma_1}{\gamma_1 - 1} z'_i + \frac{\beta}{\gamma_1 - 1} z''_i \right), \\
z'_i(A_Z^2 - A_{Z''}^2) &= \lambda^2 \left( (\gamma_1 - \gamma_2) z'_i \right), \\
z''_i(A_Z^2 - A_{Z''}^2) &= \lambda^2 \left( \frac{\alpha''\gamma_1}{\gamma_1 - 1} z_i - \frac{\beta''}{\gamma_1 - 1} z'_i + (\gamma_2 - 1) z''_i \right),
\end{align*}
\]

which in matrix notation becomes

\[
\begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix} \cdot (\hat{A}_Z^2 - \hat{A}_{Z''}^2) = \lambda^2 \begin{pmatrix} 1 - \gamma_1 & -\alpha\gamma_1 & \beta \\ 0 & \gamma_1 - \gamma_2 & 0 \\ \alpha''\gamma_1 & -\beta'' & \gamma_2 - 1 \end{pmatrix} \begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix} \triangleq \lambda^2 \cdot \Psi \cdot \begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix}.
\]

As in Condition 1 we show that \( \Psi \) is invertible. Surprisingly, we have that \( \det \Phi = -\det \Psi \), and thus Condition 1 and Condition 2 are implied by the
same requirements $\alpha''\beta \neq 9$, $\alpha'\beta'' \neq 9$, and $\alpha\beta' \neq 9$.

To prove Condition 3, first notice that the following two matrices are row-equivalent, provided that $A_{Z'} - A_Z$ is invertible.

\[
\begin{pmatrix}
I & I & I \\
A_Z & A_{Z'} & A_{Z''} \\
A_Z^2 & A_{Z'}^2 & A_{Z''}^2
\end{pmatrix}
\begin{pmatrix}
I & I & I \\
0 & I & (A_{Z'} - A_Z)^{-1}(A_{Z''} - A_Z) \\
0 & 0 & (A_Z^2 - A_{Z''}^2) - (A_Z^2 - A_{Z'}^2)(A_Z - A_{Z'})^{-1}(A_Z - A_{Z''})
\end{pmatrix}
\]

(7.34)

Therefore, Condition 3 is met if and only if the underlined matrix in (7.34) is invertible. Since $A_{Z''} - A_Z$ is invertible (given a proper choice for $\alpha, \ldots, \beta''$), it follows that the underlined matrix is invertible if and only if

\[
L \triangleq (A_Z^2 - A_{Z''}^2)(A_Z - A_{Z''})^{-1} - (A_Z^2 - A_{Z'}^2)(A_Z - A_{Z'})^{-1}
\]

is invertible. To show that $L$ is indeed invertible, we show that $\{z_i, z'_i, z''_i\} \subseteq \text{Im}(L)$ for every $i \in \{0, \ldots, \ell/3 - 1\}$. By (7.33), and by the corresponding equations for $A_Z^2 - A_{Z''}^2$, we have that

\[
\lambda^{-2}z_iL = (1 - \gamma_1)z_i(A_Z - A_{Z''})^{-1} - \frac{\alpha\gamma_1}{\gamma_1 - 1}z'_i(A_Z - A_{Z''})^{-1} + \frac{\beta}{\gamma_1 - 1}z''_i(A_Z - A_{Z''})^{-1}
\]

\[
- (1 - \gamma_2)z_i(A_Z - A_{Z'})^{-1} + \frac{\alpha\gamma_1}{\gamma_1 - 1}z'_i(A_Z - A_{Z'})^{-1} - \frac{\beta}{\gamma_1 - 1}z''_i(A_Z - A_{Z'})^{-1}
\]

\[
\lambda^{-2}z'_iL = (\gamma_1 - \gamma_2)z'_i(A_Z - A_{Z'})^{-1} + \frac{\beta'}{\gamma_1 - 1}z_i(A_Z - A_{Z'})^{-1} - (\gamma_1 - 1)z'_i(A_Z - A_{Z'})^{-1} - \frac{\alpha\gamma_1}{\gamma_1 - 1}z''_i(A_Z - A_{Z'})^{-1}
\]

\[
\lambda^{-2}z''_iL = \frac{\alpha''\gamma_1}{\gamma_1 - 1}z_i(A_Z - A_{Z''})^{-1} - \frac{\beta''}{\gamma_1 - 1}z'_i(A_Z - A_{Z''})^{-1} + (\gamma_2 - 1)z''_i(A_Z - A_{Z''})^{-1}
\]

\[- (\gamma_2 - \gamma_1)z''_i(A_Z - A_{Z'})^{-1}.
\]

193
In matrix notation, this turns to
\[
\lambda^{-2} \begin{pmatrix} z_i' \\ z_i'' \end{pmatrix} \begin{pmatrix} z_i \\
\end{pmatrix} = \begin{pmatrix}
1 - \gamma_1 & -\alpha \gamma_1 \\
\gamma_1 - \gamma_2 & 0
\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} \alpha' & \beta \\
\gamma_1 - \gamma_2 & 1 - \gamma_1 - \gamma_2
\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} \alpha'' & \beta' \\
\gamma_1 - \gamma_2 & 0
\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} (A_Z - A_{Z'})^{-1} \\
\end{pmatrix} + \begin{pmatrix}
\gamma_2 - 1 & \alpha \gamma_1 \\
\gamma_1 - \gamma_2 & 1 - \gamma_1 - \gamma_2
\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} \alpha' & \beta \\
\gamma_1 - \gamma_2 & 1 - \gamma_1 - \gamma_2
\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} (A_Z - A_{Z''})^{-1} \\
\end{pmatrix}. \quad (7.36)
\]

Now, using (7.31), and the similar equation for \( A_Z - A_{Z'} \), the expressions
\[
\begin{pmatrix} z_i \\\n\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} (A_Z - A_{Z'})^{-1} \\
\end{pmatrix}, \quad \begin{pmatrix} z_i \\\n\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} (A_Z - A_{Z''})^{-1} \\
\end{pmatrix}
\]
can be given as functions of \( z_i, z_i', z_i'', \lambda \), and the matrix \( \Phi \), e.g., by multiplying (7.31) from the right by \( (A_Z - A_{Z'})^{-1} \), and from the left by \( \lambda^{-1} \Phi^{-1} \). By performing this substitution, we have that (7.36) may be written as
\[
\lambda^{-2} \begin{pmatrix} z_i \\\n\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix} \begin{pmatrix} (A_Z - A_{Z'})^{-1} \\
\end{pmatrix} = \Upsilon \begin{pmatrix} z_i \\\n\end{pmatrix} \begin{pmatrix} z_i' \\
\end{pmatrix} \begin{pmatrix} z_i'' \\
\end{pmatrix},
\]
for some \( 3 \times 3 \) matrix \( \Upsilon \) whose entries are functions of \( 1, \gamma_1, \gamma_2, \alpha, \ldots, \beta'' \).

After some tedious calculations, we have that
\[
\det \Upsilon = \frac{\gamma_2 - 1}{-3 \lambda^3} (\alpha \alpha' \beta \beta' \beta'' + 27 \alpha \alpha' \alpha'' + 27 \beta \beta' \beta'' + 729) - 9 (\alpha' \beta \beta' \beta'' + 27 \alpha \alpha' \alpha'' + 27 \beta \beta' \beta'' + 729)
\]

We show that every possible field has a simple corresponding choice of values from \( \{1, \gamma_1, \gamma_2\} \) to \( \alpha, \ldots, \beta'' \) such that the conditions \( \det \Delta \neq 0, \alpha' \beta \neq 9, \alpha' \beta'' \neq 9, \) and \( \alpha \beta' \neq 9 \) are satisfied.

**Case 1.** If the characteristic is 2, choose \( \alpha = 1, \alpha' = \gamma_1, \alpha'' = \gamma_1, \beta = 1, \beta' = \)
\[ \gamma_2, \beta'' = 1, \text{ and then,} \]
\[ \alpha'' \beta = \gamma_1, \alpha' \beta'' = \gamma_1, \alpha \beta' = \gamma_2, \text{det } \Upsilon = \frac{622\gamma_2 - 781}{\lambda^3(-273\gamma_2 + 273)} = \frac{1}{\lambda^3(\gamma_2 + 1)} \]
and since \( 9 = 1 \notin \{\gamma_1, \gamma_2\} \), it follows that all conditions are satisfied.

**Case 2.** If the characteristic is 7, choose \( \alpha = \gamma_2, \alpha' = 1, \beta = 1, \beta' = \gamma_1, \beta'' = 1 \). Hence,
\[ \alpha'' \beta = 1, \alpha' \beta'' = 1, \alpha \beta' = 1, \text{det } \Upsilon = \frac{703(1 - \gamma_2)}{192\lambda^3(\gamma_2 + 1)} = \frac{1 - \gamma_2}{\lambda^3(\gamma_2 + 1)}, \]
and since \( 9 \neq 1 \), all conditions are met as well.

**Case 3.** If the characteristic neither 2 nor 7, choose \( \alpha = \ldots = \beta'' = 1 \), and then,
\[ \alpha'' \beta = 1, \alpha' \beta'' = 1, \alpha \beta' = 1, \text{det } \Upsilon = \frac{49(1 - \gamma_2)}{12\lambda^3(\gamma_2 + 1)}. \]
Notice that we may divide by \( 12 = 2^2 \cdot 3 \) since the characteristic is neither 2 nor 3. Since \( 9 \neq 1 \) and \( 7 \neq 0 \), all conditions are satisfied.

\[ \square \]

### 7.B.5.2 Three Parities from Two Matchings

We are now in a position to describe a construction of an \((A, S)\)-set for three parities from more than one matching.

In what follows we show that \((A, S)\)-sets which correspond to two matchings which satisfy the pairing condition, may be united to achieve a larger \((A, S)\)-set, given a proper choice of \( \lambda_x, \lambda_y \). In the following lemmas of this subsection, \( \mathcal{X} = (X, X', X''), \mathcal{Y} = (Y, Y', Y'') \) are two matchings that satisfy the pairing condition (Definition 29) and\(^9\)

\[ C_X \triangleq \{ (A_X, S_X), (A_{X'}, S_{X'}), (A_{X''}, S_{X''}) \}, \]
\[ C_Y \triangleq \{ (A_Y, S_Y), (A_{Y'}, S_{Y'}), (A_{Y''}, S_{Y''}) \}. \]

\(^9\)We omit the notations of \( \lambda_x, \lambda_y \) for convenience.


**Lemma 82** If \( C \in \{A_X, A_{X'}, A_{X''}\} \) then the eigenspace of \( C \) which corresponds to the eigenvalue \( \lambda_x \) is of the form \( \langle \{ c_0 x_i + c_1 x'_i + c_2 x''_i \}_{i=0}^{\ell/r-1} \rangle \) for some nonzero constants \( c_i's \). A similar claim holds for \( D \in \{A_Y, A_{Y'}, A_{Y''}\} \).

**Proof.** According to Lemma 79 and Lemma 81, the eigenspace of \( A_X \) which corresponds to the eigenvalue \( \lambda_x \) is \( \langle \{ 3x_i - \alpha x'_i - \beta x''_i \}_{i=0}^{\ell/3-1} \rangle \), the eigenspace of \( A'_{X} \) which corresponds to the eigenvalue \( \lambda_x \) is \( \langle \{ 3x'_i - \alpha' x''_i - \beta' x_i \}_{i=0}^{\ell/3-1} \rangle \), and the eigenspace of \( A''_{X} \) which corresponds to the eigenvalue \( \lambda_x \) is \( \langle \{ 3x''_i - \alpha'' x_i - \beta'' x'_i \}_{i=0}^{\ell/3-1} \rangle \). The claim for \( D \) is similar.

Lemma 82 together with Remark 9 and Corollary 26 implies that any eigenspace of \( C \in \{A_X, A_{X'}, A_{X''}\} \) and an eigenspace of \( D \in \{A_Y, A_{Y'}, A_{Y''}\} \) intersect at a subspace of dimension \( \ell/9 \). This gives rise to the following lemma.

**Lemma 83** Every pair of a matrix \( C \) from \( C_X \) and a matrix \( D \) from \( C_Y \) are simultaneously diagonalizable. Furthermore, \( C^2 \) and \( D^2 \) are simultaneously diagonalizable as well.

**Proof.** Let \( S_1, S_2, \) and \( S_3 \) be the eigenspaces of \( C \), let \( S_4, S_5, \) and \( S_6 \) be the eigenspaces of \( D \), and note since \( C \) and \( D \) are diagonalizable, we have that \( S_1 + S_2 + S_3 = S_4 + S_5 + S_6 = \mathbb{F}_q^\ell \). According to Corollary 26, for any \( i \in [3] \), we have that \( \dim(S_i \cap S_4) = \dim(S_i \cap S_5) = \dim(S_i \cap S_6) = \ell/9 \). Therefore, since \( S_4 + S_5 + S_6 = \mathbb{F}_q^\ell \), it follows that \( S_i \) contributes exactly \( \ell/3 \) mutual linearly independent eigenvectors of \( C \) and \( D \). Since \( S_1 + S_2 + S_3 = \mathbb{F}_q^\ell \), it follows that there exists \( \ell \) mutual linearly independent eigenvectors of \( C \) and \( D \), and hence they are mutually diagonalizable.

Now, since \( C \) and \( D \) are simultaneously diagonalizable, it follows that there exists an invertible matrix \( P \), and diagonal matrices \( E \) and \( F \), such that \( C = P^{-1}EP \) and \( D = P^{-1}FP \). This implies that \( C^2 = P^{-1}E^2P \) and \( D = P^{-1}F^2P \), and therefore, since the square of a diagonal matrix is a diagonal matrix, it follows that \( C^2 \) and \( D^2 \) are simultaneously diagonalizable as well.

**Lemma 84** If \( C \) and \( D \) are \( \ell \times \ell \) simultaneously diagonalizable matrices with no mutual eigenvalues, then \( C - D \) is invertible.

196
Proof. Let $p_0, \ldots, p_{\ell-1}$ be a basis of mutual eigenvectors of $C$ and $D$. Clearly, for all $i \in \{0, \ldots, \ell-1\}$ we have that $p_i(C - D) = p_iC - p_iD = \lambda_i p_i - \lambda_i p_i = (\lambda_i - \lambda_i) p_i$, where $\lambda_i, \lambda_i$ are the eigenvalues which correspond to $p_i$. Since $\lambda_i, \lambda_i$ are distinct, we have that $p_i \in \text{Im}(C - D)$. Therefore, since $p_0, \ldots, p_{\ell-1}$ is a basis, we have that $C - D$ is invertible.

The following lemma shows that it is possible to unite the $(A, S)$-sets $C_X, C_Y$ which were constructed using different matchings that satisfy the pairing condition (Definition 29), as long as a simple condition regarding the chosen constants $\lambda_x, \lambda_y$ is met.

Lemma 85 If $\lambda_x^6 \neq \lambda_y^6$, then $C_X \cup C_Y$ satisfies the subspace condition.

Proof. Note that the invariance, independence, and nonsingular properties which involve matrices and subspaces from one matching follow immediately from Lemma 81. It remains to prove the cases of the invariance property and the nonsingular property which involve matrices from different matchings.

To prove the invariance property, let $S \in \{S_X, S_X', S_X''\}$ and $D \in \{A_Y, A_Y', A_Y''\}$. According to Corollary 26, Remark 9, and Lemma 82, all eigenspaces of $D$ intersect with $S$ at a subspace of dimension $\ell/9$. Since the eigenspaces of $D$ are disjoint and span the entire space, it follows that $S$ contains a basis of eigenvectors of $D$. Therefore, $SD = S$. A similar proof holds if $S \in \{S_Y, S_Y', S_Y''\}$ and $D \in \{A_X, A_X', A_X''\}$.

It remains to prove Conditions 1 and 2 of the nonsingular property. To prove Condition 1 let $C \in \{A_X, A_X', A_X''\}$ and $D \in \{A_Y, A_Y', A_Y''\}$. By Lemma 83, we have that $C$ and $D$ are simultaneously diagonalizable. Moreover, according to Lemma 79, the eigenvalues of $C$ are $\lambda_x, \gamma_1 \lambda_x, \gamma_2 \lambda_x$, and the eigenvalues of $D$ are $\lambda_y, \gamma_1 \lambda_y, \gamma_2 \lambda_y$. Since $\lambda_x^6 \neq \lambda_y^6$, we have that
\[
\{\lambda_x, \gamma_1 \lambda_x, \gamma_2 \lambda_x\} \cap \{\lambda_y, \gamma_1 \lambda_y, \gamma_2 \lambda_y\} = \emptyset.
\]
Hence, Lemma 84 implies that $\text{rank}(C - D) = \ell$, which implies Condition 1 of the nonsingular property. Condition 2 also follows similarly - by Lemma 79, we have that the eigenvalues of $C^2$ are $\lambda_x^2, \gamma_1 \lambda_x^2, \gamma_2 \lambda_x^2$, and the eigenvalues of $D^2$ are $\lambda_y^2, \gamma_1 \lambda_y^2, \gamma_2 \lambda_y^2$, and since $\lambda_x^6 \neq \lambda_y^6$, it follows that $\{\lambda_x, \gamma_1 \lambda_x^2, \gamma_2 \lambda_x^2\} \cap \{\lambda_y, \gamma_1 \lambda_y^2, \gamma_2 \lambda_y^2\} = \emptyset$ (see Lemma 92 in ). Hence, Lemma 84 implies that $\text{rank}(C^2 - D^2) = \ell$, which implies Condition 2 of the nonsingular property.

As for Condition 3 let $A_i, A_j, A_k$ be matrices in the $(A, S)$-set, which
correspond to at least two distinct matchings. Recall that
\[
\begin{pmatrix}
I & I & I \\
A_i & A_j & A_t \\
A_i^2 & A_j^2 & A_t^2
\end{pmatrix}
\]
is invertible if and only if
\[
(A_i^2 - A_t^2) - (A_j^2 - A_t^2)(A_j - A_i)^{-1}(A_t - A_i)
\]
is invertible (see the proof of Lemma 81). W.l.o.g assume that \(A_i\) and \(A_j\) correspond to different matchings, and so do \(A_i\) and \(A_t\). According to Lemma 83, we have that \(A_i\) commutes with \(A_j\) and \(A_t\). Hence,
\[
\]
Multiplying from the right by \((A_t - A_i)^{-1}\), which exists by Condition 1, yields,
\[
(A_t + A_i) - (A_j + A_i) = A_t - A_j
\]
which is invertible by Condition 1.

7.B.5.3 Construction of Matchings for Three Parities

In this subsection we present a set of matchings \(\{X_i\}_{i \in [m]}\) such that any two satisfy the pairing condition, and construct the resulting \((A,S)\)-set. Recall that each vertex in the complete 3-uniform hypergraph \(K_3^\ell\) is represented by a unique unit vector of length \(\ell\). For convenience, we describe this set of matchings by considering vertex \(e_i\) as the integer \(i\) in its ternary representation. The construction of a proper set of matchings relies on the following definition, which is the three parity equivalent of Definition 31.

**Definition 33** Given an integer \(i \in [m]\) and a value \(b \in \{0,1,2\}\), the ternary cube \(C(i,b)\) is the set of all length \(m\) vectors over \(\{0,1,2\}\) that
have $b$ in entry $i$. That is,

$$ C(i, b) \triangleq \{ x \in \{0, 1, 2\}^m \mid x_i = b \}. $$

For convenience, we consider the elements in a ternary cube as ordered according to the lexicographic order (see Example 10 below).

**Example 10** If $m = 3$ then the ternary cube $C(2, 2)$ is the set \{v_1, \ldots, v_9\} such that 

$$(v_1, \ldots, v_9) = (020, 021, 022, 120, 121, 122, 220, 221, 222).$$

**Definition 34** For any $m \in \mathbb{N}$, define $m$ matchings \{X = (X_i, X'_i, X''_i)\}_{i \in [m]} as follows

$$ X_i : \begin{cases} 
X_i = C(i, 0) \\
X'_i = C(i, 1) \\
X''_i = C(i, 2). 
\end{cases} $$

**Lemma 86** The matchings from Definition 34 satisfy the pairing condition.

**Proof.** Let $X_i = (X_i, X'_i, X''_i), X_j = (X_j, X'_j, X''_j)$ be two distinct matchings. To show that these matchings satisfy the pairing condition, we must show that any edge from $X_i$ is contained in either of $X_j, X'_j, X''_j$. Let $(x_t, x'_t, x''_t)$ be an edge in $X_i$ for some $t \in [\ell/3]$. By Definition 34, we have that $x_t, x'_t, x''_t$ have the same value in every entry other than entry $i$, and $(x_t)_i = 0, (x'_t)_i = 1, (x''_t)_i = 2$. Hence, the $j$-th entry of $x_t, x'_t, x''_t$ is equal, and hence $\{x_t, x'_t, x''_t\}$ is contained in either of $X_j, X'_j, X''_j$ as required. The other direction is symmetric.

We conclude with the following theorem.

**Theorem 39** If $m$ is a positive integer, and $q$ is a prime power such that

1. if $q$ is odd, then $3|q - 1$ and $q \geq 6m + 1$,
2. if $q$ is even, then $3|q - 1$ and $q \geq 3m + 1$,

then there exists an explicitly defined $(A, S)$-set $C_1$ of size $3m$ and $3^m \times 3^m$ matrices over $\mathbb{F}_q$, which satisfies the subspace condition.
Proof. Let \( \{ X_i = (X_i, X'_i, X''_i) \}_{i \in [m]} \) be the set of matchings from Definition 34, which by Lemma 86 satisfies the pairing condition (Definition 29). Let \( \{ \lambda_i \}_{i \in [m]} \) be any set of distinct nonzero elements of \( \mathbb{F}_q \) such that \( \lambda_i^6 \neq \lambda_j^6 \) for any \( i \neq j \). Notice that the existence of such set is guaranteed in fields of either odd or even characteristic. The former is due to \( q \geq 6m+1 \), where the latter is due to the fact that \( \lambda_i^6 = \lambda_j^6 \) if and only if \( \lambda_i^3 = \pm \lambda_j^3 \), which implies \( \lambda_i^3 = \lambda_j^3 \) in fields with even characteristic, and thus \( q \geq 3m+1 \) suffices. For any \( X_i, i \in [m] \), define the code

\[
C_{X_i} = \{(A_X(\lambda_i), S_X), (A_X'(\lambda_i), S_X'), (A_X''(\lambda_i), S_X'')\}
\]

as defined in Lemma 81 and let \( C_1 = \cup_{i \in [m]} C_{X_i} \). Notice that in \( C_1 \), Condition 3 of the nonsingular property might involve matrices from three different matchings, rather than two, as was considered in the proof of Lemma 85. However, the proof of Condition 3 in Lemma 85 requires two pairs of matrices among \( \{ A_i, A_j, A_t \} \) to belong to distinct matchings, in order for the resulting \( 3 \times 3 \) matrix to be invertible. This requirement is trivially satisfied also when considering matrices from three different matchings. Hence, since the pairing condition is satisfied, and since \( \lambda_i^6 \neq \lambda_j^6 \) for all \( i \neq j \), it follows by Lemma 85 that \( C \) satisfies the subspace condition. 

7.B.6 An Improved Construction over a Larger Field

In this section, a construction with \( r = 3 \) and \( k = (r+1)m = 4m \) is presented. This construction requires a field larger than the one in Section 7.B.5, yet still linear in \( m \). This construction is comparable to [15] in terms of the parameter \( k \), but outperforms it in terms of explicitness and field size. As in [15], the construction considered in this section is not access-optimal, and is not known to achieve the sub-packetization bound. The ideas behind the construction follow the outline described in Section 7.B.3.
7.B.6.1 A Code from One Matching

A matching $\mathcal{Z} = (Z, Z', Z'')$ will provide an $(A, S)$-set of size $r + 1 = 4$, denoted by

$$(A_Z, S_Z), (A_{Z'}, S_{Z'}), (A_{Z''}, S_{Z''}).$$

As in Section 7.B.5, we assume that $3|q - 1$ in order to have three roots of unity of order 3, denoted by $1, \gamma_1, \gamma_2$. The matrices in this construction are of the form $P^{-1}AP$, where $A$ was defined in (7.29).

The matrices $P$, as in (7.15), consists of constituent $3 \times \ell$ matrices which are defined using the following operator $N$. For $u, v, w \in \mathbb{F}_q^\ell$, let

$$N(u, v, w) \triangleq \begin{pmatrix} 1 & 1 & 1 \\ 1 & \gamma_2 & \gamma_1 \\ 1 & \gamma_1 & \gamma_2 \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} u + v + w \\ u + \gamma_2 v + \gamma_1 w \\ u + \gamma_1 v + \gamma_2 w \end{pmatrix}. \quad (7.37)$$

Notice that the $3 \times \ell$ matrix $N(u, v, w)$ is row equivalent to a matrix whose rows are $u, v, w$, since $N(u, v, w)$ is defined as the multiplication of a matrix whose rows are $u, v, w$ by a Vandermonde matrix (since $\gamma_1^2 = \gamma_2$ and $\gamma_1^2 = \gamma_2$).

**Lemma 87** If $\{u_i\}_{i=0}^{\ell/3-1} \cup \{v_i\}_{i=0}^{\ell/3-1} \cup \{w_i\}_{i=0}^{\ell/3-1}$ is a basis of $\mathbb{F}_q^\ell$, then the matrix $P^{-1}AP$, where

$$P \triangleq \begin{pmatrix} N(u_0, & v_0, & w_0) \\ \vdots & \vdots & \vdots \\ N(u_{\ell/3-1}, & v_{\ell/3-1}, & w_{\ell/3-1}) \end{pmatrix},$$

has $\langle \{u_i\}_{i=0}^{\ell/3-1} \rangle$ as an eigenspace for the eigenvalue 1, $\langle \{v_i\}_{i=0}^{\ell/3-1} \rangle$ as an eigenspace for the eigenvalue $\gamma_1$, and $\langle \{w_i\}_{i=0}^{\ell/3-1} \rangle$ as an eigenspace for the eigenvalue $\gamma_2$. In addition, the subspace $\langle \{u_i + v_i + w_i\}_{i=0}^{\ell/3-1} \rangle$ is an independent subspace.

**Proof.** According to Lemma [79], the matrix $P^{-1}AP$, where the rows of $P$ are $p_0, \ldots, p_{\ell-1}$, has the following eigenspaces.
1. For the eigenvalue 1, a basis of the eigenspace is
\[ \{ p_{3i} + p_{3i+1} + p_{3i+2} \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ (u_i + v_i + w_i) + (u_i + \gamma_2 v_i + \gamma_1 w_i) + \]
\[ (u_i + \gamma_1 v_i + \gamma_2 w_i) \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ 3u_i + (1 + \gamma_1 + \gamma_2)v_i + (1 + \gamma_1 + \gamma_2)w_i \mid i \in \{0, \ldots, \ell/3-1\} \} = \{ 3u_i \}_{i=0}^{\ell/3-1} \]

2. For the eigenvalue \(\gamma_1\), a basis of the eigenspace is
\[ \{ p_{3i} + \gamma_1 p_{3i+1} + \gamma_2 p_{3i+2} \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ (u_i + v_i + w_i) + \gamma_1 (u_i + \gamma_2 v_i + \gamma_1 w_i) + \gamma_2 (u_i + \gamma_1 v_i + \gamma_2 w_i) \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ 3v_i + (1 + \gamma_1 + \gamma_2)u_i + (1 + \gamma_1 + \gamma_2)w_i \mid i \in \{0, \ldots, \ell/3-1\} \} = \{ 3v_i \}_{i=0}^{\ell/3-1} \]

3. For the eigenvalue \(\gamma_2\), a basis of the eigenspace is
\[ \{ p_{3i} + \gamma_2 p_{3i+1} + \gamma_1 p_{3i+2} \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ (u_i + v_i + w_i) + \gamma_2 (u_i + \gamma_2 v_i + \gamma_1 w_i) + \gamma_1 (u_i + \gamma_1 v_i + \gamma_2 w_i) \mid i \in \{0, \ldots, \ell/3-1\} \} = \]
\[ \{ 3w_i + (1 + \gamma_1 + \gamma_2)u_i + (1 + \gamma_1 + \gamma_2)v_i \mid i \in \{0, \ldots, \ell/3-1\} \} = \{ 3w_i \}_{i=0}^{\ell/3-1} \]

In addition, by Lemma 79 we have that \( \{ u_i + v_i + w_i \}_{i=0}^{\ell/3-1} \) is an independent subspace.

Similarly, we have the following claim about matrices of the form \( P^{-1} A^2 P \).

**Lemma 88** If \( \{ u_i \}_{i=0}^{\ell/3-1} \cup \{ v_i \}_{i=0}^{\ell/3-1} \cup \{ w_i \}_{i=0}^{\ell/3-1} \) is a basis of \( \mathbb{F}_q^\ell \), then the matrix \( P^{-1} A^2 P \) (where \( P \) was defined in Lemma 87) has \( \{ u_i \}_{i=0}^{\ell/3-1} \) as an eigenspace for the eigenvalue 1, \( \{ v_i \}_{i=0}^{\ell/3-1} \) as an eigenspace for the eigenvalue \(\gamma_1\), and \( \{ w_i \}_{i=0}^{\ell/3-1} \) as an eigenspace for the eigenvalue \(\gamma_2\). In addition, the subspace \( \{ u_i + v_i + w_i \}_{i=0}^{\ell/3-1} \) is an independent subspace.

**Proof.** According to Lemma 87 for each \( i \), \( 0 \leq i \leq \ell/3 - 1 \), we have
that
\[
\begin{align*}
    u_i \cdot P^{-1} A^2 P &= u_i \cdot (P^{-1}AP) \cdot (P^{-1}AP) \\
    &= u_i \cdot (P^{-1}AP) = u_i \\
    w_i \cdot P^{-1} A^2 P &= w_i \cdot (P^{-1}AP) \cdot (P^{-1}AP) \\
    &= \gamma_1 w_i = \gamma_1 w_i \\
    v_i \cdot P^{-1} A^2 P &= v_i \cdot (P^{-1}AP) \cdot (P^{-1}AP) \\
    &= \gamma_1 v_i = \gamma_1 v_i.
\end{align*}
\]

To see that \( S \triangleq \{u_i + v_i + w_i\}^{\ell/3-1}_{i=0} \) is an independent subspace of \( P^{-1} A^2 P \), recall that by Lemma 87, we have that \( S \) is an independent subspace of \( P^{-1}AP \), namely, \( S + S(P^{-1}AP) = \mathbb{F}_q^\ell \). Since the minimal polynomial of \( A \) is \( x^3 - 1 \), we have that \( P^{-1} A^4 P = P^{-1}AP \). Hence,
\[
S + S(P^{-1} A^2 P) + S(P^{-1} A^4 P) = S + S(P^{-1} A^2 P) + S(P^{-1} AP) = \mathbb{F}_q^\ell,
\]
and therefore \( S \) is an independent subspace of \( P^{-1} A^2 P \) as well.

Recall that the matching \( Z \) consists of the edges \( \{\{z_i, z'_i, z''_i\}\}^{\ell/3-1}_{i=0} \). The following invertible matrices are used in the construction.

\[
\begin{align*}
P_Z &\triangleq \begin{pmatrix}
    N(z_0 + z'_0 + z''_0, -z'_0, -z''_0) \\
    \vdots & \vdots & \vdots \\
    N(z_{\ell/3-1} + z'_{\ell/3-1} + z''_{\ell/3-1}, -z'_{\ell/3-1}, -z''_{\ell/3-1})
\end{pmatrix} \\
P_{Z'} &\triangleq \begin{pmatrix}
    N(-z''_0, -z_0, z_0 + z'_0 + z''_0) \\
    \vdots & \vdots & \vdots \\
    N(-z''_{\ell/3-1}, -z_{\ell/3-1}, z_{\ell/3-1} + z'_{\ell/3-1} + z''_{\ell/3-1})
\end{pmatrix} \\
P_{Z''} &\triangleq \begin{pmatrix}
    N(-z'_0, z_0 + z'_0 + z''_0, -z_0) \\
    \vdots & \vdots & \vdots \\
    N(-z'_{\ell/3-1}, z_{\ell/3-1} + z'_{\ell/3-1} + z''_{\ell/3-1}, -z_{\ell/3-1})
\end{pmatrix} \\
P_{Z'''} &\triangleq \begin{pmatrix}
    N(z_0, z'_0, z''_0) \\
    \vdots & \vdots & \vdots \\
    N(z_{\ell/3-1}, z''_{\ell/3-1}, z'_{\ell/3-1})
\end{pmatrix}
\end{align*}
\]
Definition 35 For a matching \( Z = (Z, Z', Z'') \) and any distinct nonzero field elements \( \lambda_Z, \lambda_{Z'}, \lambda_{Z''}, \) and \( \lambda_{Z^*} \), let

\[
A_Z \triangleq \lambda_Z \cdot P_Z^{-1} AP_Z, \quad S_Z \triangleq \langle Z \rangle = \left\{ z_i \right\}_{i=0}^{\ell/3-1}
\]

\[
A_{Z'} \triangleq \lambda_{Z'} \cdot P_{Z'}^{-1} AP_{Z'}, \quad S_{Z'} \triangleq \langle Z' \rangle = \left\{ z'_i \right\}_{i=0}^{\ell/3-1}
\]

\[
A_{Z''} \triangleq \lambda_{Z''} \cdot P_{Z''}^{-1} AP_{Z''}, \quad S_{Z''} \triangleq \langle Z'' \rangle = \left\{ z''_i \right\}_{i=0}^{\ell/3-1}
\]

\[
A_{Z^*} \triangleq \lambda_{Z^*} \cdot P_{Z^*}^{-1} AP_{Z^*}, \quad S_{Z^*} \triangleq \left\{ z_i + z'_i + z''_i \right\}_{i=0}^{\ell/3-1}
\]

In the following we show that in a large enough field, there exists a choice of field elements \( \lambda_Z, \lambda_{Z'}, \lambda_{Z''}, \lambda_{Z^*} \) such that the \((A, S)\)-set in Definition 35 satisfies the subspace property, and this choice can be done efficiently. Our choice will satisfy that \( \lambda_{Z'} = \lambda_Z \cdot h, \lambda_{Z''} = \lambda_Z \cdot h^2, \) and \( \lambda_{Z^*} = \lambda_Z \cdot h^3 \) for some \( h \in \mathbb{F}_q^* \). A suitable value of \( h \) and \( \lambda_Z \) will be chosen at the end of the proof of the following theorem.

Theorem 40 If \( q \) is large enough (yet independent of \( m \)), then there exists a choice of values \( \lambda_Z, \lambda_{Z'}, \lambda_{Z''}, \lambda_{Z^*} \) such that the \((A, S)\)-set from Definition 35 satisfies the subspace condition.

Proof.

By Lemma 87 we have the following table.

**Table 7.5: Eigenspaces of \( A_Z, A_{Z'}, A_{Z''}, \) and \( A_{Z^*} \).**

<table>
<thead>
<tr>
<th></th>
<th>Eigenspace for ( \lambda )</th>
<th>Eigenspace for ( \lambda_{\gamma^1} )</th>
<th>Eigenspace for ( \lambda_{\gamma^2} )</th>
<th>Independent subspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_Z, \lambda = \lambda_Z )</td>
<td>( S_{Z^*} )</td>
<td>( S_{Z'} )</td>
<td>( S_Z )</td>
<td>( S_Z )</td>
</tr>
<tr>
<td>( A_{Z'}, \lambda = \lambda_{Z'} )</td>
<td>( S_{Z^*} )</td>
<td>( S_{Z'} )</td>
<td>( S_{Z^*} )</td>
<td>( S_{Z'} )</td>
</tr>
<tr>
<td>( A_{Z''}, \lambda = \lambda_{Z''} )</td>
<td>( S_{Z^*} )</td>
<td>( S_{Z'} )</td>
<td>( S_Z )</td>
<td>( S_{Z''} )</td>
</tr>
<tr>
<td>( A_{Z^<em>}, \lambda = \lambda_{Z^</em>} )</td>
<td>( S_Z )</td>
<td>( S_{Z'} )</td>
<td>( S_Z )</td>
<td>( S_{Z^*} )</td>
</tr>
</tbody>
</table>

and hence, the independence and the invariance properties hold.

To show the nonsingular property, assume for now that \( h \) is chosen such that every distinct \( \lambda_1, \lambda_2 \) in \( \{ \lambda_Z, \lambda_{Z'}, \lambda_{Z''}, \lambda_{Z^*} \} = \{ \lambda_Z, \lambda_Z \cdot h, \lambda_Z \cdot h^2, \lambda_Z \cdot h^3 \} \) satisfy \( \lambda_1^6 \neq \lambda_2^6 \). Notice that this requirement implies that \( h^6, h^{12}, h^{18} \neq 1 \).
A specific choice of $h$ which satisfies this condition, as well as additional conditions that will emerge in the sequel, will be shown at the end of this proof.

We first show that the difference between any two matrices is of full rank. We show that $A_Z - A_{Z'}$ is of full rank, and the rest of the cases, which are similar, are given in . Notice that

$$z''_i (A_Z - A_{Z'}) = (\lambda_Z \gamma_2 - \lambda_{Z'}) z''_i$$

$$(z_i + z'_i + z''_i)(A_Z - A_{Z'}) = (\lambda_Z - \lambda_{Z'} \gamma_2)(z_i + z'_i + z''_i)$$

$$z_i (A_Z - A_{Z'}) = \lambda_Z (z_i + (1 - \gamma_1)z'_i + (1 - \gamma_2)z''_i) - \lambda_{Z'} \gamma_1 z_i$$

$$= (\lambda_Z - \lambda_{Z'} \gamma_1) z_i + \lambda_Z (1 - \gamma_1) z'_i + \lambda_Z (1 - \gamma_2) z''_i,$$

which is equivalent to

$$\begin{pmatrix} z''_i \\ z_i + z'_i + z''_i \\ z_i \end{pmatrix} \cdot (A_Z - A_{Z'}) = \lambda_Z \cdot \begin{pmatrix} 0 & 0 & \gamma_2 - h \\ 1 - h \gamma_2 & 1 - h \gamma_2 & 1 - h \gamma_2 \\ 1 - h \gamma_1 & 1 - \gamma_1 & 1 - \gamma_2 \end{pmatrix} \cdot \begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix},$$

and since

$$\begin{pmatrix} z''_i \\ z_i + z'_i + z''_i \\ z_i \end{pmatrix} = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix}$$

we may write

$$\begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix} \cdot (A_Z - A_{Z'}) \triangleq \lambda_Z \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}^{-1} \cdot \hat{\Phi}(h) \cdot \begin{pmatrix} z_i \\ z'_i \\ z''_i \end{pmatrix}. \quad (7.38)$$

To show that $z_i, z'_i, z''_i \in \text{Im}(A_Z - A_{Z'})$ it suffices to show that $\det \hat{\Phi}(h) \neq 0$. Since $\gamma_2 \neq h$ (otherwise, $h^6 = 1$), it follows that $\det \hat{\Phi}(h)$ is nonzero if and only if

$$(1 - h \gamma_2) \cdot (1 - \gamma_1) \neq (1 - h \gamma_2) \cdot (1 - h \gamma_1),$$

which also follows easily from the fact that $h^6 \neq 1$. 

205
To show that the difference between any two squares of matrices is of full rank, notice that by Lemma 88, we have the following table.

Table 7.6: Eigenspaces of $A_Z^2$, $A_Z^2$, $A_Z^2$, and $A_Z^2$.

<table>
<thead>
<tr>
<th>Eigenspace for $\lambda$</th>
<th>Eigenspace for $\lambda \gamma_1$</th>
<th>Eigenspace for $\lambda \gamma_2$</th>
<th>Independent subspace</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_Z^2$, $\lambda = \lambda_Z^2$</td>
<td>$S_{Z^*}$</td>
<td>$S_{Z''}$</td>
<td>$S_Z$</td>
</tr>
<tr>
<td>$A_Z^2$, $\lambda = \lambda_Z^2$</td>
<td>$S_{Z''}$</td>
<td>$S_{Z'}$</td>
<td>$S_{Z'}$</td>
</tr>
<tr>
<td>$A_Z^2$, $\lambda = \lambda_Z^2$</td>
<td>$S_{Z'}$</td>
<td>$S_Z$</td>
<td>$S_{Z''}$</td>
</tr>
<tr>
<td>$A_Z^2$, $\lambda = \lambda_Z^2$</td>
<td>$S_Z$</td>
<td>$S_{Z'}$</td>
<td>$S_{Z}$</td>
</tr>
</tbody>
</table>

We show that $A_Z^2 - A_Z^2$ is of full rank. The rest of the cases are similar, and are given in . Notice that

\[
\begin{align*}
  z_i''(A_Z^2 - A_Z^2) &= (\lambda_Z^2 \gamma_1 - \lambda_Z^2)z_i'' \\
  (z_i + z_i' + z_i'')(A_Z^2 - A_Z^2) &= (\lambda_Z^2 - \lambda_Z^2 \gamma_1)(z_i + z_i' + z_i'') \\
  z_i(A_Z^2 - A_Z^2) &= \lambda_Z^2(z_i + (1 - \gamma_2)z_i' + (1 - \gamma_1)z_i'') - \lambda_Z^2 \gamma_2 z_i \\
  &= \lambda_Z^2(z_i + z_i' + z_i'') - \lambda_Z^2 \gamma_2 z_i - \lambda_Z^2 \gamma_2 z_i - \lambda_Z^2 z_i,
\end{align*}
\]

which may be written as

\[
\begin{pmatrix}
  z_i'' \\
  z_i + z_i' + z_i'' \\
  z_i
\end{pmatrix} \cdot (A_Z^2 - A_Z^2) = \lambda_Z^2 \begin{pmatrix}
  0 & 0 & \gamma_1 - h^2 \\
  1 - h^2 \gamma_1 & 1 - h^2 \gamma_1 & 1 - h^2 \gamma_1 \\
  1 - h^2 \gamma_2 & 1 - \gamma_2 & 1 - \gamma_1
\end{pmatrix} \cdot \begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix} = \lambda_Z^2 \hat{\Psi}(h) \cdot \begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix},
\]

and thus, similarly, we have that

\[
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix} \cdot (A_Z^2 - A_Z^2) = \lambda_Z^2 \cdot \begin{pmatrix}
  0 & 0 & 1 \\
  1 & 1 & 1 \\
  1 & 0 & 0
\end{pmatrix}^{-1} \hat{\Psi}(h) \cdot \begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}.
\]

206
To show that \( z_i, z_i', z_i'' \in \text{Im}(A^2_Z - A^2_{Z'}) \) it suffices to show that \( \det \hat{\Psi}(h) \neq 0 \). Since \( \gamma_1 \neq h^2 \) (otherwise, \( h^6 = 1 \)) it follows that \( \det \hat{\Psi}(h) \) is nonzero if and only if

\[
(1 - h^2\gamma_1) \cdot (1 - h^2\gamma_2) \neq (1 - h^2\gamma_1) \cdot (1 - \gamma_2),
\]

which also follows easily from the fact that \( h^6 \neq 1 \).

To show that Condition 3 of the nonsingular property (i.e. that any \( 3 \times 3 \) block submatrix of the non systematic part of the generator matrix is invertible, as mentioned at the beginning of Section 7.B.5), we must show that the following matrices are invertible

\[
V(Z, Z', Z'') \triangleq \begin{pmatrix} I & I & I \\ A_Z & A_{Z'} & A_{Z''} \\ A^2_Z & A^2_{Z'} & A^2_{Z''} \end{pmatrix}, \quad V(Z, Z', Z^*) \triangleq \begin{pmatrix} I & I & I \\ A_Z & A_{Z'} & A_{Z^*} \\ A^2_Z & A^2_{Z'} & A^2_{Z^*} \end{pmatrix},
\]

\[
V(Z, Z'', Z^*) \triangleq \begin{pmatrix} I & I & I \\ A_Z & A_{Z''} & A_{Z^*} \\ A^2_Z & A^2_{Z''} & A^2_{Z^*} \end{pmatrix}, \quad V(Z', Z'', Z^*) \triangleq \begin{pmatrix} I & I & I \\ A_{Z'} & A_{Z''} & A_{Z^*} \\ A^2_{Z'} & A^2_{Z''} & A^2_{Z^*} \end{pmatrix}.
\]

Using elementary block row operations, we have that \( V(Z, Z', Z'') \) is invertible if and only if

\[
\begin{pmatrix} I & I & I \\ 0 & I & (A_{Z'} - A_Z)^{-1}(A_{Z''} - A_Z) \\ 0 & 0 & (A^2_Z - A^2_{Z''}) - (A^2_Z - A^2_{Z'}) (A_Z - A_{Z'})^{-1} (A_Z - A_{Z''}) \end{pmatrix}
\]

is invertible. Clearly, this matrix is invertible if and only if

\[
L_1 \triangleq (A^2_Z - A^2_{Z''})(A_Z - A_{Z'})^{-1} - (A^2_Z - A^2_{Z'}) (A_Z - A_{Z''})^{-1}
\]

is invertible. Similarly, \( V(Z, Z', Z^*), V(Z, Z'', Z^*), \) and \( V(Z', Z'', Z^*) \) are invertible if and only if

\[
L_2 \triangleq (A^2_Z - A^2_{Z'}) (A_Z - A_{Z'})^{-1} - (A^2_Z - A^2_{Z'}) (A_Z - A_{Z''})^{-1}
\]

\[
L_3 \triangleq (A^2_Z - A^2_{Z'}) (A_Z - A_{Z'})^{-1} - (A^2_{Z'} - A^2_{Z''}) (A_Z - A_{Z''})^{-1}
\]

\[
L_4 \triangleq (A^2_{Z'} - A^2_{Z''}) (A_{Z'} - A_{Z'})^{-1} - (A^2_{Z'} - A^2_{Z''}) (A_{Z'} - A_{Z''})^{-1}
\]
are invertible. To show that $L_1, L_2, L_3,$ and $L_4$ are invertible, we show that the image of each of them contains $z_i, z_i', z_i''$ for all $i \in \{0, \ldots, \ell/3 - 1\}$. Notice that by Table 7.6,

\[
(z_i + z_i' + z_i'')(L_1) = (z_i + z_i' + z_i'')(A_Z^2 - A_{Z''}) (A_Z - A_{Z''})^{-1} - (z_i + z_i' + z_i'')(A_Z^2 - A_{Z''}) (A_Z - A_{Z'})^{-1}
\]

\[
= (\lambda_Z^2 - \lambda_{Z''} \gamma_2) (z_i + z_i' + z_i'') (A_Z - A_{Z''})^{-1} - (\lambda_Z^2 - \lambda_{Z'} \gamma_1) (z_i + z_i' + z_i'') (A_Z - A_{Z'})^{-1}
\]

\[
z_i L_1 = z_i (A_Z^2 - A_{Z''}) (A_Z - A_{Z''})^{-1} - z_i (A_Z^2 - A_{Z'}) (A_Z - A_{Z'})^{-1} \quad (7.39)
\]

\[
= (\lambda_Z^2 (z_i + z_i' + z_i'') - \lambda_Z^2 \gamma_2 z_i' - \lambda_Z^2 \gamma_1 z_i'') (A_Z - A_{Z''})^{-1} - (\lambda_Z^2 (z_i + z_i' + z_i'') - \lambda_Z^2 \gamma_2 z_i' - \lambda_Z^2 \gamma_1 z_i'') (A_Z - A_{Z'})^{-1}
\]

\[
z_i L_1 = z_i' (A_Z^2 - A_{Z''}) (A_Z - A_{Z''})^{-1} - z_i' (A_Z^2 - A_{Z'}) (A_Z - A_{Z'})^{-1} = (\lambda_Z^2 \gamma_2 - \lambda_{Z''} \gamma_2 \gamma_1) z_i' + \lambda_Z^2 (1 - \gamma_1) z_i'' (A_Z - A_{Z'})^{-1},
\]

which can be written as

\[
\begin{pmatrix}
  z_i + z_i' + z_i'' \\
  z_i' \\
  z_i''
\end{pmatrix}
L_1 = \lambda_Z^2
\begin{pmatrix}
  1 - h^4 \gamma_2 & 1 - h^4 \gamma_2 & 1 - h^4 \gamma_2 \\
  1 - h^4 \gamma_1 & 1 - \gamma_2 & 1 - \gamma_1 \\
  0 & \gamma_2 - h^4 & 0
\end{pmatrix}
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
(A_Z - A_{Z''})^{-1}
\]

\[
+ \lambda_Z^2
\begin{pmatrix}
  1 - h^2 \gamma_1 & 1 - h^2 \gamma_2 & 1 - h^2 \gamma_1 \\
  1 - h^2 \gamma_2 & 1 - \gamma_2 & 1 - h^2 \gamma_1 \\
  h^2 (\gamma_2 - \gamma_1) & \gamma_2 - h^2 \gamma_1 & h^2 (1 - \gamma_1)
\end{pmatrix}
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
(A_Z - A_{Z'})^{-1}
\]

\[
\equiv \lambda_Z^2
\begin{pmatrix}
  U_{1,1}(h) \cdot 
  \begin{pmatrix}
    z_i \\
    z_i' \\
    z_i''
  \end{pmatrix}
  (A_Z - A_{Z''})^{-1}
\end{pmatrix}
+ U_{1,2}(h) \cdot 
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
(A_Z - A_{Z'})^{-1}
\]

Notice that the values of

\[
\begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
(A_Z - A_{Z'})^{-1}, \quad \begin{pmatrix}
  z_i \\
  z_i' \\
  z_i''
\end{pmatrix}
(A_Z - A_{Z''})^{-1}
\]

208
may easily be computed from (7.38) and the equivalent equation for \((A_Z - A_{Z''})\). Inspecting (7.38), we observe that

\[
\lambda_Z^{-1} \cdot \Phi(h)^{-1} \cdot \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \cdot \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix} = \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix} \cdot (A_Z - A_{Z''})^{-1},
\]

and hence we may write

\[
\begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix} (A_Z - A_{Z''})^{-1} = \lambda_Z^{-1} C(h)^{-1} \cdot \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix},
\]

for some matrix \(C(h)\) which depends only on \(h\). Similarly,

\[
\begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix} (A_Z - A_{Z''})^{-1} = \lambda_Z^{-1} D(h)^{-1} \cdot \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix},
\]

for some matrix \(D(h)\) which depend only on \(h\). Therefore, (7.40) can be rewritten as

\[
\begin{pmatrix} z_i + z_i' + z_i'' \\ z_i \\ z_i' \end{pmatrix} L_1 = \lambda_Z \left(U_{1,1}(h)D(h)^{-1} + U_{1,2}(h)C(h)^{-1}\right) \cdot \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix},
\]

\[
\triangleq \lambda_Z Y_1(h) \cdot \begin{pmatrix} z_i \\ z_i' \\ z_i'' \end{pmatrix},
\]

where \(Y_1(h)\) is a 3 \times 3 matrix whose entries are functions of \(h\). Note that since any entry in \(U_{1,1}(h), U_{1,2}(h), C(h)\) and \(D(h)\) is a polynomial of degree at most 6 in the variable \(h\), it follows that any entry in \(Y_1(h)\) is a rational function (that is, a division of polynomials) in \(h\), in which the degree of the enumerator and denominator polynomials is some small constant\(^{10}\). Simi-

---

\(^{10}\)This may be easily seen as a result of the well-known formula \(M^{-1} = \text{adj}M/\det M\), where \(\text{adj}M\) is the adjoint (or adjucate) matrix of \(M\).
larly, it can be shown that there exist $\Upsilon_2(h), \Upsilon_3(h), \Upsilon_4(h)$, which correspond to $L_2, L_3, L_4$, whose entries are divisions of polynomials in $h$ of small constant degree. To show that Condition 3 is satisfied, it suffices to show that $\det \Upsilon_i(h) \neq 0$ for all $i, 1 \leq i \leq 4$. A proper field constant $h$, for which non of these determinants vanish and $h^6, h^{12}, h^{18} \neq 1$, can be found by denoting

$$\det \Upsilon_i(h) = \frac{Q_{i,1}(h)}{Q_{i,2}(h)},$$

and considering the polynomial

$$Q(h) \triangleq (h^6 - 1)(h^{12} - 1)(h^{18} - 1) \prod_{i=1}^{4} (Q_{i,1}(h) \cdot Q_{i,2}(h)).$$

Clearly, if $q > \deg Q + 1$, a nonzero $h$ such that $Q(h) \neq 0$ can be found by polynomial factorization, and since $\deg Q$ is constant, the proof is complete.

This theorem showed that a single matching provides an $(A, \mathcal{S})$-set of size four, satisfying the subspace property, over a field of constant size. In the next subsection it will be shown that by taking $q$ to be at least linear in $m$, $(A, \mathcal{S})$-sets from different matchings, that satisfy the pairing condition in pairs, may be united without compromising on the subspace property.

7.B.6.2 A Code from Two Matchings

In this subsection it is shown that $(A, \mathcal{S})$-sets that were constructed from different matchings may be united, as long as the pairing condition holds.

In the remaining part of this subsection, let $\mathcal{X} = (X, X', X'')$, $\mathcal{Y} = (Y, Y', Y'')$ be two matchings which satisfy the pairing condition, and let the resulting $(A, \mathcal{S})$-sets be as in Definition 35

$$C_X \triangleq \{(A_X, S_X), (A_{X'}, S_{X'}), (A_{X''}, S_{X''}), (A_{X^*}, S_{X^*})\}$$

$$C_Y \triangleq \{(A_Y, S_Y), (A_{Y'}, S_{Y'}), (A_{Y''}, S_{Y''}), (A_{Y^*}, S_{Y^*})\}.$$  

The required values of $\lambda_X, \lambda_Y$ which are involved in the definition of these $(A, \mathcal{S})$-sets will be discussed in the sequel.

**Lemma 89** If $C \in \{A_X, A_X', A_X'', A_X^*\}$ and $D \in \{A_Y, A_Y', A_Y'', A_Y^*\}$,
then $C$ and $D$ are simultaneously diagonalizable. Furthermore, $C^2$ and $D^2$ are simultaneously diagonalizable.

**Proof.** Follow the exact outline of the proof of Lemma 88.

Recall that the definition of an $(A, S)$-set from a single matching involved the choice of two field constants $\lambda_Z$ and $h$. In what follows we use the same $h$ for all matchings, and choose proper distinct values for the constants which correspond to $\lambda_Z$. The next lemma is required for the construction.

**Lemma 90** If $\lambda_X$ and $\lambda_Y$ are two nonzero field elements such that $\lambda_Y^6 \notin \{\lambda_X^6, \lambda_X^6 h^{\pm 6}, \lambda_X^6 h^{\pm 12}, \lambda_X^6 h^{\pm 18}\}$ and

$$\begin{align*}
\lambda_1 &\in \{\lambda_X, \lambda_Y, \lambda_X^{-1}, \lambda_X^{-2}\} = \{\lambda_X, \lambda_X h, \lambda_X h^2, \lambda_X h^3\}, \\
\lambda_2 &\in \{\lambda_Y, \lambda_Y^{-1}, \lambda_Y^{-2}, \lambda_Y^{-3}\} = \{\lambda_Y, \lambda_Y h, \lambda_Y h^2, \lambda_Y h^3\},
\end{align*}$$

then

$$\begin{align*}
\{\lambda_1, \lambda_2 \gamma_1, \lambda_2 \gamma_2\} \cap \{\lambda_2, \lambda_2 \gamma_1, \lambda_2 \gamma_2\} &= \emptyset, \\
\{\lambda_1^3, \lambda_2^3 \gamma_1, \lambda_2^3 \gamma_2\} \cap \{\lambda_2^3, \lambda_2^3 \gamma_1, \lambda_2^3 \gamma_2\} &= \emptyset.
\end{align*}$$

**Proof.** We first show that $\lambda_1^6 \neq \lambda_2^6$. Assume for contradiction that $\lambda_1^6 = \lambda_2^6$. By the definition of $\lambda_1$ and $\lambda_2$, there exists $i, j \in \{0, 1, 2, 3\}$ such that $\lambda_1 = \lambda_X h^i$ and $\lambda_2 = \lambda_Y h^j$. Since $\lambda_1^6 = \lambda_2^6$, it follows that $\lambda_X^6 h^{6i} = \lambda_Y^6 h^{6j}$, and hence $\lambda_Y^6 = \lambda_X^6 h^{6(i-j)}$. Since $i, j \in \{0, 1, 2, 3\}$ it follows that $6(i-j) \in \{0, \pm 6, \pm 12, \pm 18\}$, and therefore $\lambda_Y^6 \in \{\lambda_X^6, \lambda_X^6 h^{\pm 6}, \lambda_X^6 h^{\pm 12}, \lambda_X^6 h^{\pm 18}\}$, a contradiction.

Now, if $\{\lambda_1, \lambda_2 \gamma_1, \lambda_2 \gamma_2\} \cap \{\lambda_2, \lambda_2 \gamma_1, \lambda_2 \gamma_2\} \neq \emptyset$, then there exists $i, j \in \{0, 1, 2\}$ such that $\lambda_1 \gamma_1 = \lambda_2 \gamma_1^j$, and hence, $\lambda_1^6 = \lambda_2^6$, a contradiction. Similarly, if $\{\lambda_1^3, \lambda_2^3 \gamma_1, \lambda_2^3 \gamma_1\} \cap \{\lambda_2^3, \lambda_2^3 \gamma_1, \lambda_2^3 \gamma_1\} \neq \emptyset$, there exists $i, j \in \{0, 1, 2\}$ such that $\lambda_1^3 \gamma_1 = \lambda_2^3 \gamma_1^j$, and hence, $\lambda_1^6 = \lambda_2^6$ as well, a contradiction.

Lemma 89 and Lemma 90 enable an easy choice of field elements, which induce distinct eigenvalues for the simultaneously diagonalizable matrices that correspond to distinct matchings. These distinct eigenvalues, together with the simultaneous diagonalizable matrices, will assist the proof of the following lemma, from which the construction will follow.
Lemma 91 If the field constants $\lambda_X, \lambda_Y$ satisfy

$$\lambda_Y^6 \notin \{\lambda_X^6, \lambda_X^6 h^{\pm 6}, \lambda_X^6 h^{\pm 12}, \lambda_X^6 h^{\pm 18}\},$$

then $C_X \cup C_Y$ satisfies the subspace condition.

Proof. Since each of $C_X$ and $C_Y$ satisfies the subspace condition separately, we are left to show the parts of the nonsingular property and the invariance property which involve matrices and subspaces from different matchings.

To prove the invariance property, for any $C \in \{A_X, A_X', A_{X''}, A_{X*}\}$ and any $T \in \{S_Y, S_Y', S_Y'', S_Y*\}$ we must show that $TC = T$. Let $S_1, S_2, S_3 \in \{S_X, S_X', S_{X''}, S_{X*}\}$ be the eigenspaces of $C$. It follows from Corollary 26 that $\dim(S_i \cap T) = \ell/9$ for all $i \in [3]$. Therefore, since $S_1 + S_2 + S_3 = \mathbb{F}_q^\ell$, it follows that there exists a basis $t_1, \ldots, t_{\ell/3}$ of $T$ in which all vectors are eigenvectors of $C$. Hence, for all $i \in [\ell/3]$ we have that $t_i C \in T$, and thus $TC = T$. The inverse case, where $C \in \{A_Y, A_Y', A_{Y''}, A_{Y*}\}$ and $T \in \{S_X, S_X', S_{X''}, S_{X*}\}$, is symmetric.

To prove the nonsingular property, let $C \in \{A_X, A_X', A_{X''}, A_{X*}\}$ and $D \in \{A_Y, A_Y', A_{Y''}, A_{Y*}\}$. According to Definition 35 the eigenvalues of $C$ are $\lambda_C, \lambda_C \gamma_1$, and $\lambda_C \gamma_2$ for some

$$\lambda_C \in \{\lambda_X, \lambda_X h, \lambda_X h^2, \lambda_X h^3\},$$

and similarly, the eigenvalues of $D$ are $\lambda_D, \lambda_D \gamma_1$, and $\lambda_D \gamma_2$ for some

$$\lambda_D \in \{\lambda_Y, \lambda_Y h, \lambda_Y h^2, \lambda_Y h^3\}.$$

By Lemma 90 and since $\lambda_Y^6 \notin \{\lambda_X^6, \lambda_X^6 h^{\pm 6}, \lambda_X^6 h^{\pm 12}, \lambda_X^6 h^{\pm 18}\}$, it follows that

$$\{\lambda_C, \lambda_C \gamma_1, \lambda_C \gamma_2\} \cap \{\lambda_D, \lambda_D \gamma_1, \lambda_D \gamma_2\} = \emptyset$$

$$\{\lambda_C^2, \lambda_C^2 \gamma_2, \lambda_C^2 \gamma_1\} \cap \{\lambda_D^2, \lambda_D^2 \gamma_1, \lambda_D^2 \gamma_2\} = \emptyset,$$

that is, $C$ and $D$ have no eigenvalue in common, and $C^2$ and $D^2$ have no eigenvalue in common. Since Lemma 89 implies that $C$ and $D$ are simultaneously diagonalizable, and so are $C^2$ and $D^2$, it follows by Lemma 84 that $C - D$ and $C^2 - D^2$ are invertible.

212
We are left to prove that any $3 \times 3$ block submatrix is invertible. Let $A_i, A_j, A_k$ be three matrices from $C_X \cup C_Y$ such that $A_i$ and $A_j$ are not from the same matching, and so are $A_i$ and $A_k$. Recall that, as in the proof of Theorem 40, the matrix

$$
\begin{pmatrix}
I & I & I \\
A_i & A_j & A_k \\
A_i^2 & A_j^2 & A_k^2
\end{pmatrix}
$$

is invertible if and only if

$$
L \triangleq (A_i^2 - A_k^2)(A_i - A_k)^{-1} - (A_i^2 - A_j^2)(A_i - A_j)^{-1}
$$

is invertible. Notice that by Lemma 89 $A_i$ and $A_j$ are simultaneously diagonalizable, and hence they commute. In addition, so are $A_i$ and $A_k$. Therefore,

$$
L = (A_i^2 - A_k^2)(A_i - A_k)^{-1} - (A_i^2 - A_j^2)(A_i - A_j)^{-1}
= A_i + A_k - A_i + A_j = A_k - A_j,
$$

and hence $L$ is invertible. ■

We conclude in the following theorem, in which $Q$ is the polynomial which was mentioned in the proof of Theorem 40.

**Theorem 41** If $q > \max\{42m, \deg Q\} + 1$, and $\{X_i = (X_i, X_i', X_i'')\}_{i=1}^m$ is the set of matchings from Definition 34, then the $(A, S)$-set $C_2 \triangleq \cup_{i=1}^m C_{X_i}$ satisfies the subspace condition.

**Proof.** Since $q > \deg Q + 1$, it follows that there exists $h \in \mathbb{F}_q^*$ such that $Q(h) \neq 0$. Therefore, by Theorem 40 we have that the $(A, S)$-sets $C_{X_i}$ satisfy the subspace condition separately. Since $q > 42m + 1$, the field elements $\{\lambda_{X_i}\}_{i=1}^m$ may be chosen such that every $\lambda_{X_i}, \lambda_{X_j}$ satisfy $\lambda_{X_j} \notin \{\lambda_{X_i}, \lambda_{X_i}^6, h^\pm 6, \lambda_{X_i}^6 h^\pm 12, \lambda_{X_i}^6 h^\pm 18\}$, since the choice of any $\lambda_{X_i}$ excludes the choice of at most 42 other field elements, which constitute the roots of 7 polynomials of degree 6. Notice that a proper set $\{\lambda_{X_i}\}_{i=1}^m$ may be found explicitly using a simple iterative algorithm that maintains a feasible set of elements - in each iteration it arbitrarily chooses the next element $\lambda_{X_i}$ from it.
and removes all elements $e$ which satisfy $e^6 \in \{\lambda_{X_i}^6, \lambda_{X_i}^6 h_{\pm 6}, \lambda_{X_i}^6 h_{\pm 12}, \lambda_{X_i}^6 h_{\pm 18}\}$.

Acknowledgments

The authors would like to thank Prof. Joachim Rosenthal for many helpful discussions, which contributed to this paper.
Bibliography


Appendix A

To show that $A_Z - A_{Z^*}$ is of full rank, notice that

$$z'(A_Z - A_{Z^*}) = (\lambda_Z(\gamma_1 - \lambda_Z \gamma_2))z'_i$$

$$z''(A_Z - A_{Z^*}) = (\lambda_Z \gamma_2 - \lambda_Z \gamma_1)z''_i$$

$$z_i(A_Z - A_{Z^*}) = \lambda_Z(z_i + (1 - \gamma_1)z'_i + (1 - \gamma_2)z''_i) - \lambda_{Z^*}z_i$$

$$= (\lambda_Z - \lambda_{Z^*})z_i + \lambda_Z(1 - \gamma_1)z'_i + \lambda_{Z^*}(1 - \gamma_2)z''_i$$

Since $\lambda_Z \gamma_1 \neq \lambda_Z \gamma_2$ and $\lambda_Z \gamma_2 \neq \lambda_Z \gamma_1$, we have that $z'_i, z''_i \in \text{Im}(A_Z - A_{Z^*})$. Therefore, it follows from (7.41) that $z_i \in \text{Im}(A_Z - A_{Z^*})$, since $\lambda_{Z^*} \neq \lambda_{Z^*}$.

To show that $A_Z - A_{Z''}$ is of full rank, notice that

$$z'_i(A_Z - A_{Z''}) = (\lambda_Z \gamma_1 - \lambda_{Z''})z'_i$$

$$(z_i + z'_i + z''_i)(A_Z - A_{Z''}) = (\lambda_Z - \lambda_{Z''}\gamma_1)(z_i + z'_i + z''_i)$$

$$z_i(A_Z - A_{Z''}) = \lambda_Z(z_i + (1 - \gamma_1)z'_i + (1 - \gamma_2)z''_i) - \lambda_{Z''}\gamma_2 z_i$$

$$= \lambda_Z(z_i + z'_i + z''_i) - \lambda_Z \gamma_1 z'_i - \lambda_{Z''} \gamma_2 z_i$$

Since $\lambda_Z \gamma_1 \neq \lambda_{Z''}$ and $\lambda_Z \neq \lambda_{Z''}\gamma_1$, we have that $z'_i, z_i + z'_i + z''_i \in \text{Im}(A_Z - A_{Z''})$. Therefore, it follows from (7.42) that $\lambda_Z \gamma_2 z''_i + \lambda_{Z''} \gamma_2 z_i \in \text{Im}(A_Z - A_{Z^*})$. To show that $z_i, z'_i, z''_i \in \text{Im}(A_Z - A_{Z''})$ it suffices to show that $\lambda_Z \gamma_2 \neq \lambda_{Z''} \gamma_2$, which also follows from $\lambda_Z^6 \neq \lambda_{Z''}^6$.

To show that $A_{Z'} - A_{Z''}$ is of full rank notice that

$$z_i(A_{Z'} - A_{Z''}) = (\lambda_{Z'} \gamma_1 - \lambda_{Z''} \gamma_2)z_i$$

$$(z_i + z'_i + z''_i)(A_{Z'} - A_{Z''}) = (\lambda_{Z'} \gamma_2 - \lambda_{Z''} \gamma_1)(z_i + z'_i + z''_i)$$

$$z'_i(A_{Z'} - A_{Z''}) = \lambda_{Z'}((\gamma_2 - \gamma_1)z_i + \gamma_2 z'_i + (\gamma_2 - 1)z''_i) - \lambda_{Z''} z'_i$$

$$= \lambda_{Z'} \gamma_2 (z_i + z'_i + z''_i) - \lambda_{Z'} \gamma_1 z_i - \lambda_{Z''} z''_i - \lambda_{Z''} z'_i$$

217
Therefore, \( z_i, z_i + z_i' + z_i'' \in \text{Im}(A_Z - A_{Z'}) \). Therefore, it follows from (7.43) that \( \lambda_{Z'} z_i'' + \lambda_{Z'} z_i' \in \text{Im}(A_Z - A_{Z'}) \). To show that \( z_i, z_i', z_i'' \in \text{Im}(A_Z - A_{Z'}) \) it suffices to show that \( \lambda_{Z'} \neq \lambda_{Z''} \), which also follows from \( \lambda_{Z'}^6 \neq \lambda_{Z''}^6 \).

To show that \( A_{Z'} - A_{Z''} \) is of full rank notice that

\[
\begin{align*}
z_i(A_{Z'} - A_{Z''}) &= (\lambda_{Z'} \gamma_1 - \lambda_{Z''}) z_i \\
z''(A_{Z'} - A_{Z''}) &= (\lambda_{Z'} - \lambda_{Z''} \gamma_1) z_i'' \\
z'(A_{Z'} - A_{Z''}) &= \lambda_{Z'} \left( (\gamma_2 - \gamma_1) z_i + \gamma_2 z_i' + (\gamma_2 - 1) z_i'' \right) - \lambda_{Z''} \gamma_2 z_i' \\
&= \lambda_{Z'} (\gamma_2 - \gamma_1) z_i + (\lambda_{Z'} \gamma_2 - \lambda_{Z''} \gamma_2) z_i' + \lambda_{Z''} (\gamma_2 - 1) z_i''
\end{align*}
\]

Since \( \lambda_{Z'} \gamma_1 \neq \lambda_{Z''} \) and \( \lambda_{Z'} \neq \lambda_{Z''} \gamma_1 \), we have that \( z_i, z_i'' \in \text{Im}(A_{Z'} - A_{Z'}) \). Therefore, it follows from (7.44) that \( z_i' \in \text{Im}(A_{Z'} - A_{Z'}) \), since \( \lambda_{Z'} \gamma_2 \neq \lambda_{Z''} \gamma_2 \).

To show that \( A_{Z''} - A_{Z'} \) is of full rank notice that

\[
\begin{align*}
z_i(A_{Z''} - A_{Z'}) &= (\lambda_{Z''} \gamma_2 - \lambda_{Z'}) z_i \\
z'(A_{Z''} - A_{Z'}) &= (\lambda_{Z''} - \lambda_{Z'} \gamma_2) z_i' \\
z''(A_{Z''} - A_{Z'}) &= \lambda_{Z''} \left( (\gamma_1 - \gamma_2) z_i + (\gamma_1 - 1) z_i' + \gamma_1 z_i'' \right) - \lambda_{Z'} \gamma_2 z_i' \\
&= \lambda_{Z''} (\gamma_1 - \gamma_2) z_i + \lambda_{Z''} (\gamma_1 - 1) z_i' + (\lambda_{Z''} \gamma_1 - \lambda_{Z'} \gamma_1) z_i''
\end{align*}
\]

Since \( \lambda_{Z''} \gamma_2 \neq \lambda_{Z'} \) and \( \lambda_{Z''} \neq \lambda_{Z'} \gamma_2 \), we have that \( z_i, z_i' \in \text{Im}(A_{Z''} - A_{Z'}) \). Therefore, it follows from (7.45) that \( z_i' \in \text{Im}(A_{Z''} - A_{Z'}) \), since \( \lambda_{Z''} \gamma_1 \neq \lambda_{Z'} \gamma_1 \).

**Appendix B**

To show that \( A_{Z,2}^2 - A_{Z,2}^2 \) is of full rank, notice that

\[
\begin{align*}
z_i'(A_{Z,2}^2 - A_{Z,2}^2) &= (\lambda_{Z,2}^2 \gamma_2 - \lambda_{Z,2}^2 \gamma_1) z_i' \\
z_i''(A_{Z,2}^2 - A_{Z,2}^2) &= (\lambda_{Z,2}^2 \gamma_1 - \lambda_{Z,2}^2 \gamma_2) z_i'' \\
z_i(A_{Z,2}^2 - A_{Z,2}^2) &= \lambda_{Z,2}^2 (z_i + (1 - \gamma_2) z_i' + (1 - \gamma_1) z_i'') - \lambda_{Z,2}^2 z_i \\
&= (\lambda_{Z,2}^2 - \lambda_{Z,2}^2 \gamma_2) z_i + \lambda_{Z,2}^2 (1 - \gamma_2) z_i' + \lambda_{Z,2}^2 (1 - \gamma_1) z_i''
\end{align*}
\]

Since \( \lambda_{Z,2}^2 \gamma_2 \neq \lambda_{Z,2}^2 \gamma_1 \) and \( \lambda_{Z,2}^2 \gamma_1 \neq \lambda_{Z,2}^2 \gamma_2 \), it follows that \( z_i', z_i'' \in \text{Im}(A_{Z,2}^2 - A_{Z,2}^2) \). Therefore, (7.46) implies that \( z_i \in \text{Im}(A_{Z,2}^2 - A_{Z,2}^2) \), since \( \lambda_{Z,2}^2 \neq \lambda_{Z,2}^2 \).
To show that $A_{Z}^2 - A_{Z''}^2$, is of full rank notice that

\begin{align*}
  z_i'(A_{Z}^2 - A_{Z''}^2) &= (\lambda_{Z}^2 \gamma_2 - \lambda_{Z''}^2 \gamma_1) z_i' \\
  (z_i + z_i' + z_i'')(A_{Z}^2 - A_{Z''}^2) &= (\lambda_{Z}^2 \gamma_1 - \lambda_{Z''}^2 \gamma_2)(z_i + z_i' + z_i'') \\
  z_i(A_{Z}^2 - A_{Z''}^2) &= \lambda_{Z}^2 (z_i + (1 - \gamma_2) z_i' + (1 - \gamma_1) z_i'') - \lambda_{Z''}^2 \gamma_1 z_i \\
  &= \lambda_{Z}^2 (z_i + z_i' + z_i'') - \lambda_{Z}^2 \gamma_2 z_i' - \lambda_{Z''}^2 \gamma_1 z_i \\
  &\neq \lambda_{Z}^2 \gamma_2 \neq \lambda_{Z''}^2 \gamma_2, \text{it follows that } z_i', z_i + z_i' + z_i'' \in \text{Im}(A_{Z}^2 - A_{Z''}^2). \text{ Therefore, (7.47) implies that } \lambda_{Z}^2 \gamma_1 z_i'' + \lambda_{Z''}^2 \gamma_2 z_i \in \text{Im}(A_{Z}^2 - A_{Z''}^2). \\
  \text{Hence, to have that } z_i, z_i', z_i'' \in \text{Im}(A_{Z}^2 - A_{Z''}^2), \text{ it suffices to show that } \lambda_{Z}^2 \gamma_1 \neq \lambda_{Z''}^2 \gamma_2, \text{ which is implied by } \lambda_{Z}^6 \neq \lambda_{Z''}^6.
\end{align*}

To show that $A_{Z'}^2 - A_{Z''}^2$, is of full rank notice that

\begin{align*}
  z_i'(A_{Z'}^2 - A_{Z''}^2) &= (\lambda_{Z'}^2 \gamma_2 - \lambda_{Z''}^2 \gamma_1) z_i \\
  (z_i + z_i' + z_i'')(A_{Z'}^2 - A_{Z''}^2) &= (\lambda_{Z'}^2 \gamma_1 - \lambda_{Z''}^2 \gamma_2)(z_i + z_i' + z_i'') \\
  z_i'(A_{Z'}^2 - A_{Z''}^2) &= \lambda_{Z'}^2 ((\gamma_1 - \gamma_2) z_i + \gamma_1 z_i' + (\gamma_1 - 1) z_i'') - \lambda_{Z''}^2 z_i' \\
  &= \lambda_{Z'}^2 \gamma_1 (z_i + z_i' + z_i'') - \lambda_{Z'}^2 \gamma_2 z_i - \lambda_{Z''}^2 z_i' \\
  &\neq \lambda_{Z'}^2 \gamma_2 \neq \lambda_{Z''}^2 \gamma_2, \text{it follows that } z_i, z_i + z_i' + z_i'' \in \text{Im}(A_{Z'}^2 - A_{Z''}^2). \text{ Therefore, (7.48) implies that } \lambda_{Z'}^2 z_i'' + \lambda_{Z''}^2 m_i' \in \text{Im}(A_{Z'}^2 - A_{Z''}^2). \text{ Hence, to have that } m_i, m_i', m_i'' \in \text{Im}(A_{Z'}^2 - A_{Z''}^2), \text{ it suffices to show that } \lambda_{Z'}^2 \neq \lambda_{Z''}^2, \text{ which is implied by } \lambda_{Z}^6 \neq \lambda_{Z''}^6.
\end{align*}

To show that $A_{Z'}^2 - A_{Z''}^*$, is of full rank notice that

\begin{align*}
  z_i(A_{Z'}^2 - A_{Z''}^*) &= (\lambda_{Z'}^2 \gamma_2 - \lambda_{Z''}^2 \gamma_1) z_i \\
  z_i''(A_{Z'}^2 - A_{Z''}^*) &= (\lambda_{Z'}^2 - \lambda_{Z''}^2 \gamma_2) z_i'' \\
  z_i'(A_{Z'}^2 - A_{Z''}^*) &= \lambda_{Z'}^2 ((\gamma_1 - \gamma_2) z_i + \gamma_1 z_i' + (\gamma_1 - 1) z_i'') - \lambda_{Z''}^2 \gamma_1 z_i' \\
  &= \lambda_{Z'}^2 \gamma_1 (z_i + z_i' + z_i'') - \lambda_{Z'}^2 \gamma_2 z_i - \lambda_{Z''}^2 \gamma_1 z_i' \\
  &\neq \lambda_{Z'}^2 \gamma_2 \neq \lambda_{Z'}^2 \gamma_2, \text{it follows that } z_i, z_i' \in \text{Im}(A_{Z'}^2 - A_{Z''}^*). \text{ Therefore, (7.49) implies that } m_i' \in \text{Im}(A_{Z'}^2 - A_{Z''}^*), \text{ since } \lambda_{Z'}^2 \gamma_1 \neq \lambda_{Z''}^2 \gamma_1.
\end{align*}
To show that $A_{Z''} - A_{Z'}$ is of full rank notice that
\[
\begin{align*}
    z_i(A_{Z''} - A_{Z'}) &= (\lambda_{Z''}^2 \gamma_1 - \lambda_{Z'}^2)z_i \\
    z'_i(A_{Z''} - A_{Z'}) &= (\lambda_{Z''}^2 - \lambda_{Z'}^2 \gamma_1)z'_i \\
    z''_i(A_{Z''} - A_{Z'}) &= \lambda_{Z''}^2 \left( (\gamma_2 - \gamma_1)z_i + (\gamma_2 - 1)z'_i + \gamma_2 z''_i \right) - \lambda_{Z'}^2 \gamma_2 z''_i \\
    &= \lambda_{Z''}^2 (\gamma_2 - \gamma_1)z_i + \lambda_{Z''}^2 (\gamma_2 - 1)z'_i + (\lambda_{Z''}^2 \gamma_2 - \lambda_{Z'}^2 \gamma_2)z''_i
\end{align*}
\]
Since $\lambda_{Z''}^2 \gamma_1 \neq \lambda_{Z'}^2$, and $\lambda_{Z''}^2 \neq \lambda_{Z'}^2 \gamma_1$, it follows that $z_i, z'_i \in \text{Im}(A_{Z''}^2 - A_{Z'}^2)$. Therefore, (7.50) implies that $z''_i \in \text{Im}(A_{Z''}^2 - A_{Z'}^2)$, since $\lambda_{Z''}^2 \gamma_2 \neq \lambda_{Z'}^2 \gamma_2$.

Appendix C

Lemma 92 For nonzero constants $\lambda_x, \lambda_y$ in $\mathbb{F}_q$, if $\lambda_x^6 \neq \lambda_y^6$ then
\[
\{\lambda_x, \gamma_1 \lambda_x, \gamma_2 \lambda_x\} \cap \{\lambda_y, \gamma_1 \lambda_y, \gamma_2 \lambda_y\} = \emptyset.
\]

Proof. For any $i, j \in [3]$, let $\lambda_x \gamma_1^i \in \{\lambda_x, \gamma_1 \lambda_x, \gamma_2 \lambda_x\}$ and $\lambda_y \gamma_1^j \in \{\lambda_y, \gamma_1 \lambda_y, \gamma_2 \lambda_y\}$. Since $\gamma_1$ is a root of unity of order 3, if $\lambda_x \gamma_1^i = \lambda_y \gamma_1^j$, then $(\lambda_x \gamma_1^i)^6 = (\lambda_y \gamma_1^j)^6$, and hence $\lambda_x^6 = \lambda_y^6$, a contradiction. \[\blacksquare\]
Chapter 8

Coding for Locality in Reconstructing Permutations

Netanel Raviv, Eitan Yaakobi, and Muriel Médard

8.A Conference Version

Abstract

The problem of storing permutations in a distributed manner arises in several common scenarios, such as efficient updates of a large, encrypted, or compressed data set. This problem may be addressed in either a combinatorial or a coding approach. The former approach boils down to presenting large sets of permutations with locality, that is, any symbol of the permutation can be computed from a small set of other symbols. In the latter approach, a permutation may be coded in order to achieve locality. This paper focuses on the combinatorial approach.

We provide upper and lower bounds for the maximal size of a set of permutations with locality, and provide several simple constructions which attain the upper bound. In cases where the upper bound is not attained, we provide alternative constructions using Reed-Solomon codes, permutation polynomials, and multi-permutations.
8.A.1 Introduction

For an integer $n$, let $S_n$ be the group of all permutations on $n$ elements. Given a permutation $\pi \in S_n$ we consider the problem of storing a representation of $\pi$ in a distributed system of storage nodes. This problem arises when considering efficient permutation updates to a distributed storage system. That is, in a system which stores a file with large entries whose order commonly changes, one might prefer to store the permutation of the entries, rather than constantly shift them around. Alternatively, the stored file may be signed, hashed, or compressed, and storing the permutation alongside the file allows to update the file without altering its signature. Perhaps the most natural example for an update is the common operation of cut and paste, which may be modeled as a permutation update.

The crux of enabling efficient storage lies in the notion of locality, that is, any failed storage node may be reconstructed by accessing a small number of its neighbors. The corresponding coding problem is often referred to as symbol locality, in which every symbol of a codeword is a function of a small set of other symbols.

In this paper we consider symbol locality. Further, since our underlying motivation is allowing small updates to be done efficiently, we disregard the notion of minimum distance between the stored permutations, and focus solely on locality.

Locality in permutations may be considered in either a combinatorial or a coding approach. Under the combinatorial approach, which is the main one in this paper, the underlying motivation is set aside, and the problem boils down to finding (or bounding the maximum size of) sets of permutations which present locality. Under the coding approach, the given permutation may be coded in order to achieve locality, e.g. by using a locally recoverable code (LRC). The combinatorial approach clearly outperforms the use of LRCs in terms of redundancy (see Section 8.A.2), at the price of not being able to store any permutation. Furthermore, it may be shown [10] that storing a subset of $S_n$ using an LRC while maintaining the same overhead as in the combinatorial approach does not enable an instant access to the elements of the permutation, as discussed further in this section.

The combinatorial approach may also be applied in rank modulation coding for flash memories [9], in which each flash cell contains an electric
charge, and a block of cells contains the permutation which is induced by the charge levels. A rank modulation code which enables local erasure correction allows quick recovery from a complete loss of charge in a cell. Yet, this application requires some further adjustments of our techniques, since the charge levels usually represent relative values rather than absolute ones.

A system which stores \( \pi \in S_n \) is required to answer either \( \pi^{-1}(i) = ? \) (denoted Q1) or \( \pi(i) = ? \) (denoted Q2) quickly, for any \( i \). In the combinatorial approach, either one of Q1 or Q2 becomes trivial, depending if we consider the permutation at hand as \((\pi(1), \ldots, \pi(n))\) or \((\pi^{-1}(1), \ldots, \pi^{-1}(n))\). That is, when storing the latter, answering Q1 is straightforward, and answering Q2 is possible by inspecting \( \pi^{-1}(i), \pi^{-1}(\pi^{-1}(i)), \ldots, \) etc., until \( i \) is found (see [6, ch. 1.3, p. 29]). Hence, the number of required queries for Q1 is 1 (or \( \log n \) bits), and for Q2 it is at most the length of the longest cycle in \( \pi \). Although it is not the general purpose of this research, we take initial steps towards efficient retrieval of \( \pi(i) \) and \( \pi^{-1}(i) \) simultaneously. A more expansive discussion will appear in the full version of this paper.

Since a variety of mathematical techniques are used throughout this paper, in each technique we consider the permutations in \( S_n \) as operating on a different sets of symbols. These sets may be either \([n] \triangleq \{1, \ldots, n\}\) or \(\{0, \ldots, n-1\}\). Alternatively, we may assume that \( n \) is a power of prime, and \(\{0,1,\ldots,n-1\}\) is an enumeration of the elements in \( \mathbb{F}_n \), the finite field with \( n \) elements, where the additive identity element of \( \mathbb{F}_n \) is denoted by “0” and the multiplicative identity element is denoted by “1”. Unless otherwise stated, we consider permutations in the one line representation (one-liner, in short), that is, \( \pi \triangleq (\pi_1, \ldots, \pi_n) = (\pi^{-1}(1), \ldots, \pi^{-1}(n)) \). Given a set \( S \subseteq S_n \), we say that \( S \) has locality \( d \) if for any \( \pi \in S \), any symbol \( \pi_i \) may be computed from \( d \) other symbols of \( \pi \). The rate of \( S \) is defined as \( \log |S| / \log(n! \).

This paper is organized as follows. Section 8.A.2 summarizes related previous work. Section 8.B.3 discusses upper and (existential) lower bounds on the maximal possible size of subsets of \( S_n \) which present locality. Section 8.A.4 provides several simple constructions, some of which attain the upper bound presented in Section 8.A.3. One of these constructions is enhanced by using Reed-Solomon codes and permutation polynomials in Subsection 8.A.4.3, and by using multi-permutations in Section 8.A.4.4. Concluding remarks and problems for future research are given in Section 8.A.6.
For the lack of space, some proofs are omitted, and are included in the full version of this paper [10]. Additional omitted results are briefly summarized in Section 8.A.5.

8.A.2 Previous Work

Coding over $S_n$, endowed with either of several possible metrics [5], was extensively studied under many different motivations. For example, codes in $S_n$ under the Kendall’s $\tau$ metric [1] and the infinity metric [13] were shown to be useful for non-volatile memories, and codes under the Hamming metric (also known as permutation arrays) were shown to be useful for power-line communication [4]. In all of these works, the permutations are encodings of messages, and hence should maintain minimum distance constraints. In this work, however, the permutation itself is of interest, and thus minimum distance is not considered.

As mentioned in the Introduction, we consider permutations in their one line representation (one-liner, in short). Our problem may be seen as allowing local erasure correction of permutations in the one-liner. Erasure and deletion correction of permutation codes was discussed in [8]. In this work it was shown that the most suitable metric for erasure correction (called “stable erasure” in [8]) is the Hamming metric, that measures the number of entries in which the one-liners differ. However, the work of [8] was motivated by the rank modulation scheme in flash memories and thus locality was not discussed.

Furthermore, it is obvious that a permutation array with minimum Hamming distance $n - d + 1$ allows local erasure correction of any symbol from any $d$ other symbols. However, constructing permutation arrays with minimum Hamming distance is an infamously hard problem, let alone in the high distance region [2]. Moreover, construction of permutation arrays with minimum Hamming distance is not equivalent to finding sets of permutations with locality, since the inverse is clearly untrue, that is, a set with locality $d$ does not imply a permutation array with minimum Hamming distance $n - d + 1$.

A similar motivation lies in the work of [11], where the authors considered updates which involve deletions and insertions to a file in a distributed storage system. Clearly, a permutation update can be seen as a series of
deletions and insertions and conversely, a deletion is treated in [11] as a permutation. Our work may be seen as an extension of “scheme P” from [11] to permutation updates, as we handle various types of larger sets of permutations.

When considering the coding approach, a standard technique is to use LRCs. An \((m,k,d)\) LRC is a code that produces an \(m\)-symbol codeword from a \(k\)-symbol message, such that any symbol of the produced codeword may be recovered by contacting at most \(d\) other symbols. LRCs have been subject to extensive research in recent years [12], mainly due to their application in distributed storage systems. Consider any permutation \(\pi \in S_n\) as a string over the alphabet \([n]\), and encode it to \(m\) symbols using an optimal systematic LRC. LRCs that encode \(n = k\) symbols to \(m\) symbols and admit locality of \(d\) satisfy [12, Theorem 2.1]

\[
\frac{n}{m} \leq \frac{d}{d+1},
\]

i.e., their rate is bounded from above by \(d/(d+1)\). Thus, \(n/d\) redundant information symbols are required to achieve locality of \(d\). Using the combinatorial approach we achieve smaller storage overhead, in the price of not being able to store any permutation. In addition, in Subsection 8.A.3.2 it will be shown that there exists a coset of an optimal locally recoverable code \(C\), which contains a set \(S\) of words that can be considered as permutations. However, this claim is merely existential, and does not provide any significant insights on the structure of \(S\).

### 8.A.3 Bounds

Let \(A(n, d)\) be the maximum size of a subset of \(S_n\) with locality \(d\). This section presents an upper bound and an existential lower bound on \(A(n, d)\). This upper bound is later improved for \(d = 1\), and is attained by a certain construction in Section 8.A.4.1 to follow.

\[1\]More precisely, the alphabet \([n]\) when seen as a subset of a large enough finite field \(F_q\), over whom the construction of the LRC is possible.
8.A.3.1 Upper Bounds

The bound for LRCs (8.1) can be used as-is if \( n \) is a power of prime, and the set of permutations is considered as a non-linear code in \( \mathbb{F}_n \). By a simple adaptation of [12, Theorem 2.1] to non-linear codes, we have that a non-linear code in \( \mathbb{F}_n \) with locality \( d \) contains at most \( n^{\lfloor dn/(d+1) \rfloor} \) codewords. This bound may be improved by utilizing the combinatorial structure of permutations.

**Theorem 42** \( A(n, d) \leq \frac{n!}{\frac{d+1}{d^d}} \).

Using the Stirling approximation, Theorem 42 implies an upper bound of \( \frac{d^d}{d} \) on the rate of a set of permutations with locality \( d \).

The trivial subset \( C = S_n \) admits locality of \( d = n - 1 \), and attains the upper bound. In addition, the alternating group, and its complement, have locality of \( n - 2 \). This is due to the fact that a given permutation with two erased symbols can be corrected to either of two possible permutations, one of which is odd and the other is even. Hence, the alternating group and its complement attain this upper bound as well. According to these examples, we have that \( A(n, n-1) = n! \), and \( A(n, n-2) = n!/2 \).

For \( d < n - 2 \) there exists a large gap between this bound and the sizes of the sets presented in this paper. This gap may be resolved for \( d = 1 \) by using a graph theoretic argument on the dependency graph in the proof of Theorem 42.

As a result, we obtain the following bound on the maximal size of sets of permutations with locality one.

**Theorem 43** \( A(n, 1) \leq n!! \triangleq \prod_{i=0}^{\lceil n/2 \rceil - 1} (n - 2i) \).

Since the set constructed in Section 8.A.4.1 below attains the bound of Theorem 43 for \( d = 1 \), we have that \( A(n, 1) = n!! \).

8.A.3.2 Lower Bound

Optimal LRC of length \( n \) and locality \( d \) may easily be constructed over \( \mathbb{Z}_n \), the set of integers modulo \( n \). This is done by adding \( n/(d+1) \) “parity checks” to all disjoint sets of \( d \) consecutive symbols in \( \mathbb{Z}_n^{-n/(d+1)} \). This requires that \( d+1 \) divides \( n \), but may easily be adapted to any \( d \). The rate
of this code attains the upper bound of \( \frac{n-n/(d+1)}{n} = \frac{d}{d+1} \), given in (8.1), and since the code is linear, all its cosets have locality \( d \) as well. Since \( n! \) of the words in \( \mathbb{Z}^n \) are permutations, we obtain the following existential lower bound on \( A(n,d) \).

**Theorem 44** \( A(n,d) \geq \frac{n!}{n^{n/(d+1)}} \).

The rate which is implied by Theorem 44 asymptotically attains the rate of the upper bound which is implied by Theorem 42. Yet, the upper and lower bounds do not coincide, since Theorem 48 implies higher redundancy (that is, \( \log(n!) - \log|S| \)) than the one implied by Theorem 46. It is evident from Theorem 46 and Theorem 48 that enabling larger locality may potentially increase the size of the sets.

### 8.A.4 High Rate Constructions

This section presents several constructions of sets of permutations with locality, some of which attain the upper bound given in Section 8.A.3.1. The first set of permutations, discussed in Section 8.A.4.1, is those that may be seen as a concatenation of \( n/h \) permutations in \( S_h \), for some \( h \) which divides \( n \). Subsection 8.A.4.2 shows a similar technique which achieves high locality. Subsection 8.A.4.3 and Subsection 8.A.4.4 enhance the construction of Subsection 8.A.4.1 by using Reed-Solomon codes over permutation polynomials, and by using multi-permutations.

#### 8.A.4.1 Concatenation of Short Permutations

Obviously, in the one-line representation, any single symbol may easily be computed from all other symbols. This principle leads to simple sets of permutations which can be stored efficiently.

Consider the set \( S \) of permutations in \( S_n \) which may be viewed as a concatenation of \( n/h \) shorter permutations on \( h \) elements, for some integer \( h \) which divides \( n \). That is, their one-liner may be viewed as a concatenation of \( n/h \) one-liners, each of which is a permutation of either of the sets \( \{1, \ldots, h\}, \{h + 1, \ldots, 2h\}, \) etc. Clearly, \( S \) contains \( (h!)^{n/h} \cdot (n/h)! \) permutations, has locality \( d = h - 1 \) and rate \( \frac{1}{d+1} \).
Note that multiple erasures can be corrected simultaneously, as long as they do not reside in the same short permutation. Two erasures from the same short permutation cannot be corrected simultaneously. In addition, Q1 can be answered trivially, and Q2 requires finding the suitable sub-permutation in $n/h$ queries, and additional $h$ queries to locate the desired element.

For $d = 1$ we have $|S| = n!!$, and thus this construction attains the bound of Theorem 43 with equality. However, for any $d = O(1)$, $d \geq 2$, these sets do not attain the optimal rate, and are superseded by the existential lower bound of Theorem 44.

8.A.4.2 Concatenation of Range-Restricted Permutations

In this subsection we provide a technique for producing sets of permutations with high locality $d \geq n/2$. For a set of symbols $\Sigma$ let $S(\Sigma)$ denote the set of all permutations of $\Sigma$. In this subsection we use the alphabet $\Sigma = \{0, \ldots, n-1\}$, and hence $S(\Sigma) = S_n$. Let $h$ be an integer which divides $n$, and for $i \in \{0, \ldots, n/h - 1\}$ let

$$K_i \triangleq S\{ih, ih + 1, \ldots, (i+1)h - 1\} \circ S([n] \setminus \{ih, ih + 1, \ldots, (i+1)h - 1\}),$$

where $\circ$ denoted the ordinary concatenation of sequences.

**Lemma 93** The set $S \triangleq \bigcup_{i=0}^{n/h-1} K_i$ has locality $d = n - h - 1$.

**Proof.** To repair a missing symbol $\pi_j, 0 \leq j \leq n - 1$ in $\pi \in S$, distinguish between the cases $j \leq h - 1$ and $j \geq h$. If $j \leq h - 1$, $\pi_j$ may clearly be computed from $\{\pi_i\}_{i \in \{0, \ldots, h-1\}\setminus\{j\}}$. If $j \geq h$, the set of symbols $\{\pi_i\}_{i \in \{h, \ldots, n-1\}\setminus\{j\}}$ must contain a gap of $h$ consecutive numbers, which are located in the prefix of $\pi$. After identifying this gap, the missing symbol $\pi_j$ may easily be deduced. ■

The set $S$ contains $\frac{n}{h} \cdot h! \cdot (n-h)! = n \cdot (h-1)! \cdot (n-h)!$ and it does not attain the upper bound given in Theorem 42. For constant $h$ the rate of $S$ asymptotically approaches 1 as $n$ goes to infinity, since

$$\frac{\log(n \cdot (h-1)! \cdot (n-h)!)}{\log(n!)} \geq \frac{\log((n-h)!) - \log(n-1)!}{\log(n!)} \xrightarrow{n \to \infty} 1.$$
Equal rate may be obtained for lower locality, where \( h = \Theta(n) \); if \( h = \delta n \) for some constant \( 0 < \delta < 1 \), then

\[
\frac{\log(n \cdot (h - 1)! \cdot (n - h)!)}{\log(n!)} \xrightarrow{n \to \infty} \delta + (1 - \delta) = 1.
\]

An identical rate is also obtained by choosing \( h = \Theta(n^\epsilon) \). Hence, the best choice of parameters for this technique seems to be \( h = \Theta(n) \), since it results in low locality and optimal rate.

8.A.4.3 Extended Construction from Error-Correcting Codes

This section provides a construction of a set of permutations in \( S_n \) with locality, from two constituent ingredients. The first ingredient is a set of permutations \( S \subseteq S_{n-t} \) with locality \( d \), for some given \( t \) and \( d \). The second ingredient is an error-correcting code \( T \), in which all codewords consist of \( t \) distinct symbols.

A **symbol replacement function** \( f \) is an injective function which maps one alphabet to another. Given a permutation \( \pi \) and a symbol replacement function \( f \) let \( f(\pi) \) be the result of replacing the symbols of \( \pi \) according to \( f \). For a set of permutations \( S \) let \( f(S) \triangleq \{ f(\pi) | \pi \in S \} \). The construction of this section relies on the following observation.

**Observation 4** If \( S \subseteq S_{n-t} \) is a set of permutations with locality \( d \), and \( f \) is a symbol replacement function, then \( f(S) \) is a set of permutations with locality \( d \) as well.

Using a proper symbol replacement function \( f \), a permutation \( f(\pi) \) for \( \pi \in S \) is concatenated to a codeword from \( T \) to create a permutation in \( S_n \). This symbol replacement function is given in the following definition, which is followed by an example.

**Definition 36** For any integers \( 1 < t < n \), let \( \pi \) be a permutation in \( S_{n-t} \) and \( e \in [n]^t \) be a word with \( t \) distinct symbols \( \{ \sigma_1, \ldots, \sigma_t \} \triangleq E \subseteq [n] \). Let
Let \( f_E \) be the following symbol replacement function

\[
f_E : [n-t] \rightarrow ([n-t] \setminus E) \cup \{n-t+1, \ldots, n-t+|E \cap [n-t]|\}
\]

\[
f_E(i) = \begin{cases} 
  i, & i \notin E. \\
  j, & \text{For some integer } s, \text{ } i \text{ and } j \text{ are the } s\text{-smallest numbers in } E \cap [n-t] \text{ and } \\
  & \{n-t+1, \ldots, n\} \setminus E, \text{ respectively.}
\end{cases}
\]

That is, \( f_E \) maps each element which does not appear in \( E \) to itself, and each element which appears in \( E \) is mapped to a symbol in \( \{n-t+1, \ldots, n\} \) which does not appear in \( E \), in an increasing manner. Using \( f_E \), define the operator \( \circ \) as

\[
\pi \circ e \triangleq f_E(\pi) \circ e,
\]

where \( \circ \) denotes the ordinary concatenation of strings.

**Example 11** For \( n = 7 \) and \( t = 3 \), let \( \pi = (1,2,3,4) \), \( e = (3,4,7) \), and \( E = \{3,4,7\} \). By Definition [36] we have that

\[
f_E(1) = 1, \ f_E(2) = 2, \ f_E(3) = 5, \ f_E(4) = 6, \ \text{and} \ \pi \circ e = f_E(\pi) \circ e = (1,2,5,6,3,4,7) \in S_7.
\]

The operation \( \circ \) is used to extend an existing set \( S \subseteq S_{n-t} \) with locality to a subset of \( S_n \) with a larger locality by using an error-correcting MDS code \( T \).

**Lemma 94** For integers \( 1 < t < n \), if \( S \subseteq S_{n-t} \) is a set with locality \( d \) and \( T \) is an MDS code in \([n]^t\) with minimum distance \( \delta \) and distinct symbols, then \( S \circ T \triangleq \{s \circ e | s \in S, \ e \in T\} \subseteq S_n \) is a set of permutations with locality \( d + t - \delta + 1 \).

**Proof.** Let \( \pi = s \circ e \) be a permutation in \( S \circ T \). To repair a missing symbol \( \pi_j \) for \( 1 \leq j \leq n \) we distinguish between the cases \( j \leq n-t \) and \( j > n-t \). If \( j > n-t \), by the minimum distance property of the MDS code \( T \) we may obtain \( \pi_j \) by accessing \( t - \delta + 1 \) symbols from \( e \). If \( j \leq n-t \), then
by accessing $t - \delta + 1$ symbols from $e$ we may identify the function $f_E$ used to define the operator $\odot$ (Definition 36). Once $f_E$ is known, the symbol $\pi_j$ may be obtained by using Observation 4.

This technique can be used to obtain explicit sets with constant locality $d \geq 2$, which are the largest ones in this paper for this locality. Unfortunately, to the best of our knowledge the asymptotic rate of these sets does not exceed $\frac{1}{2}$, and hence they are not optimal. Moreover, since a set with locality 1 also has locality $d \geq 2$ for any $d$, the sets of locality 1 from Subsection 8.A.4.1 can be used for any locality greater than 1, while obtaining rate of $\frac{1}{2}$ as well. Nevertheless, for small values of $d$ we are able to construct explicit sets with locality $d$ which contain more permutations than the sets with locality 1 from Subsection 8.B.4.2 To provide good examples by this technique, we must construct error-correcting codes where each codeword consists of distinct symbols.

Recall that a Reed-Solomon code is given by evaluations of degree restricted polynomials on a fixed set of distinct elements from a large enough finite field. These codes contain sub-codes which are suitable for our purpose. The codewords in these sub-codes are obtained by evaluations of permutation polynomials. A permutation polynomial is a polynomial which represents an injective function from $\mathbb{F}_n$ to itself. In spite of the very limited knowledge on permutation polynomials in general, all permutation polynomials of degree at most 5 are known (see [4, Table 2]). For example, we have the following lemma.

**Lemma 95** [4, Table 2] If $n$ is a power of 2, then there exist at least $(n - 1)(2n + \frac{n(n^2 + 2)}{3})$ permutation polynomials of degree at most 4 over $\mathbb{F}_n$.

As a corollary, we obtain the following constructions.

**Example 12** Let $n$ be an integer power of 2, and let $S \subseteq S_{n-6}$ be an optimal set with locality 1 (which exists by Subsection 8.A.4.1 since $n-6$ is even). Let $T$ be a subset of a Reed-Solomon code of dimension 5 and length 6 over $\mathbb{F}_n$, which corresponds to all permutation polynomials of degree at most 4. According to Lemma 94 and Lemma 95 the set $B \triangleq S \odot T$ contains $(n - 6)!! \cdot (n - 1)(2n + \frac{n(n^2 + 2)}{3})$ permutations, and has locality 6.

Notice that an optimal set $A \subseteq S_n$ with locality 1, which may be seen as having any larger locality, contains $n!!$ permutations (see Section 8.A.4.1).
The set $B$ is larger, since $n!! = (n - 6)!! \cdot \Theta(n^3)$ and $|B| = (n - 6)!! \cdot \Theta(n^4)$. Hence, Example 12 provides sets which are at least $n$ times larger than those given in Section 8.A.4.1 and have larger constant locality. Additional examples are provided in [10].

8.A.4.4 High-Locality Construction From Multi-Permutations

While constructing sets of permutations with constant locality $d \geq 2$ and rate above $\frac{1}{2}$ seems hard, it is fairly easy to construct sets with such rate and locality $d = \Theta(n^\epsilon)$, for $0 \leq \epsilon \leq 1$. Such a set is obtained from Section 8.A.4.1 by taking $h = \Theta(n^\epsilon)$. However, the resulting rate is $\epsilon$, where Theorem 44 guarantees that for this locality there exist sets with rate which tends to 1 as $n$ tends to infinity.

In this subsection it is shown that the construction from Section 8.A.4.1 may be enhanced by using multi-permutations, achieving rate of $\frac{1}{2} + \frac{\epsilon}{2}$ for locality $d = \Theta(n^\epsilon)$. The methods and notations in this subsection are strongly based on [3].

For nonnegative integers $\ell$ and $m$, a balanced multi-set $\{1^m, 2^m, \ldots, \ell^m\}$ is a collection of the elements in $[\ell]$, where each element appears $m$ times. A multi-permutation on a balanced multi-set is a string of length $\ell m$, which is given by a function $\sigma : [\ell m] \rightarrow [\ell]$ such that for all $i \in [\ell]$, $|\{j | \sigma(j) = i\}| = m$. The set of all multi-permutations is denoted by $S_{\ell,m}$, and its size is $\frac{(\ell m)!}{(m!)^\ell}$. To distinguish between different appearances of the same element in a multi-permutation $\sigma$, for $j \in [\ell m]$, $i \in [\ell]$, and $r \in [m]$ we denote $\sigma(j) = i_r$ and $\sigma^{-1}(i_r) = j$ if the $j$-th position of $\sigma$ contains the $r$-th appearance of $i$.

**Example 13** If $m = 2$ and $\ell = 3$ then $\pi = (1,1,2,3,2,3)$ is a multi-permutation on the balanced multi-set $\{1,1,2,3,2,3\}$. To refer to the second appearance of 2 we say that $\pi(5) = 2_2$.

We are interested in multi-permutations with two appearances of each element, and therefore assume that $m = 2$ and $\ell = n/2$. In particular, we consider such multi-permutations in which any two appearances of the same element are not too far apart. To this end, the following definition is required.
Definition 37 If \( \pi \in S_{n/2,2} \) and \( t \in [n] \) then,

\[
w(\pi) \triangleq \max_{i \in [n/2]} |\pi^{-1}(i_1) - \pi^{-1}(i_2)| , \quad \text{and} \quad B_t \triangleq \{ \pi \in S_{n/2,2} : w(\pi) \leq t \}.
\]

That is, \( w(\pi) \) indicates the maximum distance between two appearances of the same element, or alternatively, \( w(\pi) - 1 \) indicates the maximum number of elements between two appearances of the same element in \( \pi \). For a given \( t \), \( B_t \) is the set of all multi-permutations in \( S_{n/2,2} \) in which every two identical elements are separated by at most \( t - 1 \) other elements. Clearly, the multi-permutation \( \pi \) which was given in Example 13 is in \( B_2 \).

To construct “ordinary” permutations in \( S_n \) from multi-permutations in \( S_{n/2,2} \) we use the term assignment of permutations. As in Subsection 8.A.4.2 for a set of elements \( \Sigma \) we denote by \( S(\Sigma) \) the set of all permutations of \( \Sigma \) (that is, the set of all injective functions \( f : \{1, \ldots, |\Sigma|\} \rightarrow \Sigma \)).

Definition 38 If \( \pi \in S_{n/2,2} \) and \( \gamma_1, \ldots, \gamma_{n/2} \) are permutations such that \( \gamma_i \in S(\{2i-1, 2i\}) \) for all \( i \), then \( \sigma = \pi(\gamma_1, \ldots, \gamma_{n/2}) \) is the permutation in \( S_n \) such that for all \( 1 \leq j \leq n \), if \( \pi(j) = i_r \) then \( \sigma(j) = \gamma_i(r) \).

Example 14 If \( \pi = (1, 1, 2, 3, 2, 3) \in S_{3,2} \) and \( \gamma_1 = (1, 2) \), \( \gamma_2 = (4, 3) \), and \( \gamma_3 = (6, 5) \) then \( \sigma = \pi(\gamma_1, \gamma_2, \gamma_3) = (1, 2, 4, 6, 3, 5) \in S_6 \).

Note that by choosing \( h = 2 \) in the construction which appears in Subsection 8.A.4.1 for the resulting set \( S \) can be described as

\[
S = \{ \pi(\gamma_1, \ldots, \gamma_{n/2}) \mid \forall i, \gamma_i \in S(\{2i-1, 2i\}) \text{ and } \pi \in B_1 \}.
\]

Hence, the construction in the following lemma may be seen as a generalization of the construction from Subsection 8.B.4.2.

Lemma 96 For a nonnegative integer \( t \), the set

\[
A_t \triangleq \{ \pi(\gamma_1, \ldots, \gamma_{n/2}) \mid \forall i, \gamma_i \in S(\{2i-1, 2i\}), \text{ and } \pi \in B_t \}
\]

has locality \( 4t \).

Using this lemma, we are able to provide a set with high locality \( \Theta(n^\epsilon) \), and asymptotic rate strictly above \( \frac{1}{2} \).
Theorem 45 If \( t = \Theta(n^\epsilon) \) then \( \lim_{n \to \infty} \frac{\log |A_t|}{\log n^n} \geq \frac{1}{2} + \frac{\epsilon}{2} \).

8.A.5 Additional Results

Due to space constraints, some of the results from the full version of this paper were omitted. We list some of the omitted results below, and the interested reader may find them, together with full proofs of all the included results, in [10].

For certain low values of locality, a lower bound equivalent to Theorem 44 is obtained by a connection to a classic problem in combinatorics. This problem is known as the toroidal semi-queens problem, or alternatively, a set of transversals in a cyclic Latin square [7]. It can be shown that given an efficient algorithm which produces transversals in a cyclic Latin square, one may construct a linear set of permutations with locality and optimal rate. However, such algorithm does not currently exists, and in fact, an estimation of the number of transversals in cyclic Latin squares was only recently given in [7].

As mentioned in Section 8.A.2 in this paper the permutations themselves are of interest, as opposed to most of the research in permutation codes, where the permutations are a means to overcome technical limitations. For this reason we seek insightful structures of permutations which induce locality, and not necessarily provide a non-vanishing rate.

One such structure is given by a ball in the infinity metric on \( S_n \), i.e., the set of permutations in which every element is located no more than \( r \) positions from its original location, for some given radius \( r \). These permutations arise naturally in scenarios where an initial conjectured ranking of items is imposed, and any item is not expected to exceed its initial ranking by more than a certain bound. For a given radius \( r \), we show in [10] that the corresponding permutations have locality of \( 4r \), and concurrent erasures may be handled simultaneously more efficiently than separately. Although the exact size of the ball in the infinity metric is not known, it is known to be exponential.

Another interesting structure arises in consumption of media, where the consumer begins with an arbitrary item of a feed, and either proceeds forward or backwards from the set of consecutive items which he read so far. This procedure induces \( 2^n - 2 \) permutations in which any prefix (or suffix)
consists of consecutive numbers, and admits locality of (at most) four.

In this paper we discussed the storage problem of permutations from a combinatorial point of view, with no encoding. Needless to say that this restriction, albeit being mathematically appealing, is merely a narrow interpretation of the wide spectrum of techniques which can be devised to store permutations in a distributed manner. In the full paper, we take several initial steps towards expanding our arsenal by allowing encoding (“the coding approach”). In this approach we show that a ball in the infinity metric admits a more efficient representation with the same locality. Additionally, we present a framework for supporting queries of arbitrary powers of the stored permutation, a technique which is interconnected with the combinatorial approach. We conclude with a proof of concept that permutations can be stored with less redundancy than ordinary strings, achieving a (highly) negligible advantage for locality of two and three.

8.A.6 Discussion and Open Problems

In this paper we discussed locality in permutations without any encoding, motivated by applications in distributed storage and rank modulation codes. The lack of encoding enables to maintain low query complexity, which is a reasonable requirement in our context. Clearly, if no such constraint is assumed, any permutation can be represented using \( \lceil \log(n!) \rceil \) bits, and stored using an LRC. However, when a query complexity requirement is imposed, there seems to be much more to be studied, and our results are hardly adequate comparing with the potential possibilities. Additional discussion about techniques which involve encoding appears in the full version of this paper.

We provided upper and lower bounds for the maximal size of a set of permutations with locality, and provided several simple constructions with high rate. For simplicity, we assumed that each node stores a single symbol from \([n]\), and focused on symbol locality. This convention may be adjusted to achieve storage systems with different parameters, i.e., one might impose an array code structure on this problem, in order to improve the parameters.

Finally, we list herein a few specific open problems which were left unanswered in this work.
1. Close the gap between the upper bound in Theorem 42 and the lower bound in Theorem 44 potentially by using the methods of Theorem 47.

2. Provide an explicit construction of sets with constant locality \( d \geq 2 \) and optimal rate \( \frac{d}{d+1} \). The existence of these sets is guaranteed by Theorem 44.

3. Find additional large sets of permutations that have good locality.

4. Explore the locality of permutations under different representation techniques.

5. Endow \( S_n \) with one of many possible metrics, and explore the locality of codes with a good minimum distance by this metric.

Acknowledgments

The work of Netanel Raviv was supported in part by the Aharon and Ephraim Katzir study grant, the IBM Ph.D. fellowship, and the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant no. 10/12. The authors would like to thank Ahmad Beirami for fruitful discussions.
Bibliography


8.B Unpublished Full Version

Abstract

The problem of storing permutations in a distributed manner arises in several common scenarios, such as efficient updates of a large, encrypted, or compressed data set. This problem may be addressed in either a combinatorial or a coding approach. The former approach boils down to presenting large sets of permutations with locality, that is, any symbol of the permutation can be computed from a small set of other symbols. In the latter approach, a permutation may be coded in order to achieve locality. Both approaches must present low query complexity to allow the user to find an element efficiently. We discuss both approaches, and give a particular focus to the combinatorial one.

In the combinatorial approach, we provide upper and lower bounds for the maximal size of a set of permutations with locality, and provide several simple constructions which attain the upper bound. In cases where the upper bound is not attained, we provide alternative constructions using a variety of tools, such as Reed-Solomon codes, permutation polynomials, and multi-permutations. In addition, several low-rate constructions of particular interest are discussed.

In the coding approach we discuss an alternative representation of permutations, present a paradigm for supporting arbitrary powers of the stored permutation, and conclude with a proof of concept that permutations may be stored more efficiently than ordinary strings over the same alphabet.

8.B.1 Introduction

For an integer \( n \), let \( S_n \) be the group of all permutations on \( n \) elements. Given a permutation \( \pi \in S_n \) we consider the problem of storing a representation of \( \pi \) in a distributed system of storage nodes. This problem arises when considering efficient permutation updates to a distributed storage system. For example, in a system which stores large entries whose order commonly changes, one might prefer to store the permutation of the entries, rather than constantly shift them around. Alternatively, the stored file may be signed or hashed (using cryptographic primitives), and storing the per-
mutation alongside the file allows to update the file without altering its signature. The given file may also be compressed using a source code, and storing the permutation enables to perform permutation updates without the need to decompress the file. Perhaps the most natural example for such a scenario is the common operation of cut and paste, which may be modelled as a permutation update, and briefly discussed in Subsection 8.B.1.1.

Above all questions of efficiency, in very simple storage schemes, a permutation update might require to decode the file (see Subsection 8.B.1.1). Storing the permutation alongside x allows a permutation update to be made without decoding the file, at the price of storage overhead.

The crux of enabling efficient storage lies in the notion of locality, that is, any failed storage node may be reconstructed by accessing a small number of its neighbors. The corresponding coding problem is often referred to as symbol locality, in which every symbol of a codeword is a function of a small set of other symbols [25]. Another approach towards efficient storage stems from array codes, in which each storage node stores a large set of symbols from the codeword (e.g., [27] and references therein). For simplicity, in this paper we consider symbol locality. Furthermore, since our underlying motivation is allowing small updates to be done efficiently, we disregard the notion of minimum distance between the stored permutations, and focus solely on locality. Occasionally, we will discuss local correction of simultaneous erasures, as in [22].

As mentioned earlier, locality in permutations may be considered in either a combinatorial or a coding approach. Under the combinatorial one, the underlying motivation is set aside, and the problem boils down to finding (or bounding the maximum size of) sets of permutations which present locality. This approach is the main one in this paper. Under the coding approach, the given permutation may be coded in order to achieve locality, e.g., by using a locally recoverable code (LRC). The combinatorial approach clearly outperforms the use of LRCs in terms of redundancy (see Section 8.B.2), at the price of not being able to store any permutation. Furthermore, it is also shown in Section 8.B.2 that storing a subset of \( S_n \) using an LRC while maintaining the same overhead as in the combinatorial approach does not enable an instant access to the elements of the permutation, as discussed further in this section.

The combinatorial approach may also be applied in rank modulation
coding for flash memories \cite{16}, in which each flash cell contains an electric charge, and a block of cells contains the permutation which is induced by the charge levels. Flash memories are susceptible to various types of hardware failures, some of which result in a complete loss of the charge in a cell, i.e., an erasure \cite{15}. A rank modulation code which enables local erasure correction allows quick recovery from a such loss of charge. Yet, this application requires some further adjustments of our techniques, since the charge levels usually represent relative values rather than absolute ones.

Several natural questions, which are irrelevant in ordinary storage, may arise when discussing storage of permutations. For example, a storage system which stores \( \pi \in S_n \) may be required to answer either \( \pi^{-1}(i) =? \) or \( \pi(i) =? \) quickly. These questions are denoted by Q1 and Q2, respectively, and notice that without this additional requirement, storing permutations reduces to storing binary strings of length \( \lceil \log(n!) \rceil \) by enumerative encoding. In the combinatorial approach, either one of Q1 or Q2 becomes trivial, depending if we consider the permutation at hand as \((\pi(1), \ldots, \pi(n))\) or \((\pi^{-1}(1), \ldots, \pi^{-1}(n))\). For example, when storing the latter, answering Q1 is straightforward, and answering Q2 is possible by inspecting \( \pi^{-1}(i), \pi^{-1}(\pi^{-1}(i)), \ldots \), etc., until \( i \) is found (see \cite{12} ch. 1.3, p. 29). Hence, the number of required queries for Q1 is 1 (or \( \log n \) bits), and for Q2 it is at most the length of the longest cycle in \( \pi \). Although it is not the general purpose of this research, we take initial steps towards efficient retrieval of \( \pi(i) \) and \( \pi^{-1}(i) \) simultaneously. Clearly, allowing the permutation to be encoded provides more freedom in devising storage techniques. However, maintaining a concise representation which enables Q1 and Q2 to be answered quickly is a rather involved question, which was studied in the past in a non-distributed setting (e.g. \cite{20,21}, see further details in Section 8.B.2).

Since a variety of mathematical techniques are used throughout this paper, in each technique we consider the permutations in \( S_n \) as operating on a different sets of symbols. These sets may be either \([n] \triangleq \{1, \ldots, n\}\) or \(\{0, \ldots, n-1\}\). Alternatively, we may assume that \( n \) is a power of prime, and \(\{0,1,\ldots,n-1\}\) is an enumeration of the elements in \( \mathbb{F}_n \), the finite field with \( n \) elements, where the additive identity element of \( \mathbb{F}_n \) is denoted by “0” and the multiplicative identity element is denoted by “1”. Unless otherwise stated, we consider permutations in the one line representation (one-liner,
in short), that is, \( \pi \triangleq (\pi_1, \ldots, \pi_n) = (\pi^{-1}(1), \ldots, \pi^{-1}(n)) \). Given a set \( S \subseteq S_n \), we say that \( S \) has locality \( d \) if for any \( \pi \in S \), any symbol \( \pi_i \) may be computed from \( d \) other symbols of \( \pi \). The rate of \( S \) is defined as \( \log |S|/\log(n!) \), the identity permutation is denoted by \( \text{Id} \), and \( \circ \) denotes the concatenation of sequences.

This paper is organized as follows. Additional motivation for studying storage of permutations is given in Subsection 8.B.1.1. Section 8.B.2 summarizes related previous work. Section 8.B.3 discusses upper and (existential) lower bounds on the maximal possible size of subsets of \( S_n \) which present locality. Section 8.B.4 provides several simple constructions, some of which attain the upper bound which is presented in Section 8.B.3. A construction which shows a connection to Reed-Solomon codes via permutation polynomials, and a construction via multi-permutations are also given in Section 8.B.4. In Section 8.B.5 two sets of permutations of particular interest are presented. These sets have low rate, and low locality. The set from Subsection 8.B.5.1 will be later shown to have a more efficient representation (Subsection 8.B.6.1). The coding approach is discussed in Section 8.B.6, in which the main result is a technique which allows to compute every power of the stored permutation very efficiently, and is strongly based on [20]. Subsection 8.B.6.3 shows a preliminary proof of concept that permutations may be stored using less redundancy bits than ordinary strings, and shows a concrete technique of doing so, attaining a negligible advantage. Concluding remarks and problems for future research are given in Section 8.B.7.

8.B.1.1 Motivation

This subsection presents a general motivation for distributed storage of permutations through common file updates, and through applications in cryptography. We begin by showing that in a certain simple scenario, permuting the coordinates of the stored file without decoding is impossible.

Assume that a file \( x \), which contains \( n \) entries, is divided into two halves \( x_1, x_2 \) and stored in three nodes using the simplest parity check code \( x_1, x_2, x_1 + x_2 \) (as in the RAID4 storage system). In addition, assume that the user would like to apply \( \pi \in S_{n/2} \) on \( x_1 \). Applying \( \pi \) only over the systematic part, that is, update the system to contain \( \pi(x_1), x_2, x_1 + x_2 \), clearly does not maintain the error correction capability. On the other hand, up-
dating the parity node to contain $\pi(x_1) + x_2$ without knowing either $x_1$ or $x_2$ is information theoretically impossible. Hence, given $\pi$, the storage node which contains $x_1 + x_2$ cannot update its own content to $\pi(x_1) + x_2$. This fact is illustrated in the following lemma.

**Lemma 97** If $x_1$ and $x_2$ are strings of length $n/2$ over a field $\mathbb{F}_q$, then given a non-trivial $\pi \in S_{n/2}$ and $x_1 + x_2$, it is information theoretically impossible to compute $\pi(x_1) + x_2$.

**Proof.** It is widely known that given $x_1 + x_2$, one cannot infer any information on either of $x_1$ and $x_2$. We show that if $\pi(x_1) + x_2$ may be computed, then some information about certain symbols of $x_1$ can be inferred.

Knowing $x_1 + x_2$ and $\pi(x_1) + x_2$, we may calculate $\alpha \triangleq \pi(x_1) + x_2 - (x_1 + x_2) = \pi(x_1) - x_1$. Furthermore, since $\pi \in S_{n/2}$ is non-trivial, we may arbitrarily choose one cycle $(i_1, i_2, \ldots, i_t)$ from the disjoint cycle representation of $\pi$, for some $t > 1$. Hence, we may assemble the following linear system of equations, in the variables $x_{i_1}^{i_1}, x_{i_1}^{i_2}, \ldots, x_{i_1}^{i_t}$.

\[
\begin{pmatrix}
-1 & 1 & 0 & 0 & \cdots & 0 \\
1 & -1 & 0 & 0 & \cdots & 0 \\
0 & 1 & -1 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 1 & -1 & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
x_{i_1}^{i_1} \\
x_{i_1}^{i_2} \\
\vdots \\
x_{i_1}^{i_t}
\end{pmatrix}
= 
\begin{pmatrix}
\alpha_{i_1} \\
\alpha_{i_2} \\
\vdots \\
\alpha_{i_t}
\end{pmatrix}
\] (8.2)

Since the matrix in (8.2) has rank $t - 1 > 0$, and since the system has a solution, we have that the affine space of solutions of this system is of dimension 1. Therefore, the sequence $x_{i_1}^{i_1}, \ldots, x_{i_1}^{i_t}$ may have either one of $q$ possible values, rather than $q^t$ values. Since $t > 1$, the claim follows.

Permutation updates arise in everyday scenarios. Two very common updates may be modeled as a special case of permutation updates. One is the well-known *cut-paste* operation, in which a portion of data is removed and placed elsewhere in the file. The other one is the *replacement* operation, in which two distinct portions of equal size switch places. The following simple lemmas present the structure and number of the permutations of these two operations.

243
Lemma 98  A cut-paste update is a permutation update whose corresponding permutation is a cyclic shift. In particular, if a portion of $\alpha$ consecutive symbols, beginning at position $s$ in the file, is being moved to position $t$, the corresponding permutation is a cycle of length $t - s + 1$, which performs a cyclic shift of $\alpha$ spots to the elements between positions $s$ and $t$.

It should be noted that for $\alpha = 1$ and $s \neq t$, Lemma 98 describes a set of permutations called translocations. Translocations are the building blocks of the Ulam metric [14, Proposition 3], which is an essential tool for error-correction in flash memories [14] and DNA research [5, 11].

Lemma 99  A replacement update is a permutation update whose corresponding permutation is a product of disjoint transpositions.

Transpositions can be seen as the building blocks of the transposition metric, considered in [17]. The number of cut-paste and replacement operations is rather small, as shown in the following lemma, which is easy to prove.

Lemma 100  The number of replacement updates on a file of $n$ elements is $O(n^3)$, and the number of cut-paste operations on a file of $n$ elements is $O(n^3)$.

Clearly, storing one of $O(n^3)$ possible values requires $O(\log n)$ bits. Therefore, providing an efficient answer to Q1 and Q2 becomes trivial, since obtaining all information about one of these permutations is possible by reading $O(\log n)$ bits. Thus, these specific permutations, however common, will not be discussed further in this paper. Yet, other interesting types of permutations arise in daily scenarios, see Section 8.B.5.

Similar questions were studied from a cryptographic perspective. The works of [3, 4] initiated the research of cryptographic primitives, such as hash functions or signature schemes, that enable efficient updates. That is, small changes in the file may be incorporated into its hash (or signature) efficiently, without requiring to recompute it from scratch. The updates considered in [3, 4] are of different nature (replacement rather than permutations), and the file is not stored in a distributed manner, yet the underlying motivation is highly similar, that is, how to perform efficient updates to a stored file, without the need to encode it anew.
Our work may be used in conjunction with any existing (distributed) cryptosystem, which was not necessarily meant for updates, while enabling permutation updates to be done efficiently. Furthermore, in coding-based cryptosystems (such as the McEliece cryptosystem \[15\]) any update of the un-encrypted file is inefficient. This is since any change of a symbol in the un-encrypted file requires changing at least as many symbols of its encrypted counterpart as the minimum distance of the underlying code.

### 8.B.2 Previous Work

Coding over $S_n$, endowed with either of several possible metrics \[11\], was extensively studied under many different motivations. For example, codes in $S_n$ under the Kendall’s $\tau$ metric \[2\] and the infinity metric \[26\] were shown to be useful for non-volatile memories, and codes under the Hamming metric (also known as permutation arrays) were shown to be useful for power-line communication \[9\]. In all of these works, the permutations are encodings of messages, and hence should maintain minimum distance constraints. In this work, however, the permutation itself is of interest, thus minimum distance is not considered, and certain sets of permutations with low rate are also of interest. In addition, when taking the coding approach, the added redundancy need not to comply with any combinatorial constraints, e.g., to result in a permutation of a larger base set.

As mentioned in the Introduction, we consider permutations in their one line representation (one-liner, in short). Our problem may be seen as allowing local erasure correction of permutations in the one-liner. Erasure and deletion correction of permutation codes was discussed in \[15\]. In this work it was shown that the most suitable metric for erasure correction (called “stable erasure” in \[15\]) is the Hamming metric, that measures the number of entries in which the one-liners differ. However, the work of \[15\] was motivated by the rank modulation scheme in flash memories and thus locality was not discussed.

Furthermore, it is obvious that a permutation array with minimum Hamming distance $n - d + 1$ allows local erasure correction of any symbol from any $d$ other symbols. However, constructing permutation arrays with minimum Hamming distance is an infamously hard problem, let alone in the high distance region \[6\]. Moreover, construction of permutation arrays with min-
imum Hamming distance is not equivalent to finding sets of permutations with locality, since the inverse is clearly untrue, that is, a set with locality \( d \) does not imply a permutation array with minimum Hamming distance \( n - d + 1 \).

A similar motivation lies behind the work of [23], where the authors considered updates of a distributed storage system which involve deletions and insertions to a file which is stored in a distributed system. Clearly, a permutation update can be seen as a series of deletions and insertions. The so-called “scheme P” [23 Section 4.2] provides a framework for maintaining a file \( x \in \mathbb{F}_q^n \) in a distributed storage system under insertions and deletions. The entire file is assumed to be stored using an arbitrary array-code, and the deletions and insertions are taking place with respect to any specific block. Interestingly, a deletion is treated as a permutation, where the deleted symbol is replaced with the symbol 0 and pushed to the end of the block. A set of permutation matrices \( \{A^{(i)}\} \), one for each block, is stored in the system to keep track of the permuted symbols. An insertion is treated similarly, keeping track of the location of insertion using the permutation matrices. The overhead of storing the matrices \( \{A^{(i)}\}_{i \in [n]} \) is improved by using the fact that the entire matrices need not to be stored, and we may settle for the locations of the edits. Our work may be seen as an extension of scheme P from [23] to permutation updates, as we handle various types of larger sets of permutations.

Recall that we are interested in supporting the queries Q1 or Q2. A similar problem, which relates to general strings rather than to permutations, was addressed in the past as a problem about data structures. A representation technique called “the succincter”, which enables one-symbol recovery,

\[2\]

was discussed in [21]. This representation requires only slightly more bits than the optimal one, but it seems to be superfluous when discussing large alphabets. Formally, according to [21 Theorem 1], for any \( t \), a file \( x \) of length \( n \) over an alphabet \( \Sigma \) can be represented by \( O(|\Sigma| \log n) + f(n, x, t) \) for some function \( f \), while allowing one-symbol recovery in \( O(t) \) time. In our scenario we have that \( \Sigma = [n] \), and optimal one-symbol recovery is trivially possible by using an \( n \log n \) bit representation.

When considering the coding approach, a standard technique may be

\[2\] One-symbol recovery is the term used in [21] for returning a given entry of the stored string, without requiring to decode it.
the use of *Locally Recoverable Codes* (LRCs). An \((m, k, d)\) LRC is a code that produces an \(m\)-symbol codeword from a \(k\)-symbol message, such that any symbol of the produced codeword may be recovered by contacting at most \(d\) other symbols. LRCs have been subject to extensive research in recent years [25], mainly due to their application in distributed storage systems. Consider any permutation \(\pi \in S_n\) as a string over the alphabet \([n]\), and encode it to \(m\) symbols using an optimal systematic LRC (e.g., [25]). Singleton-optimal LRCs that encode \(n = k\) symbols to \(m\) symbols and admit locality of \(d\) must satisfy \(q \geq m\), and [25, Theorem 2.1]

\[
\frac{n}{m} \leq \frac{d}{d + 1},
\]

i.e., their rate is bounded from above by \(d/(d + 1)\). Thus, \(n/d\) redundant information symbols are required to achieve locality of \(d\). Using the combinatorial approach (and some of the techniques in the coding approach, see Section 8.B.6.1) we achieve smaller storage overhead, in the price of not being able to store any permutation. Notice that by restricting our attention to a subset \(S \subseteq S_n\) it is possible to obtain locality of \(d\) (assuming each storage node may contain \(\log n\) bits) by using LRCs with \(\log |S| \cdot \frac{d+1}{d}\) bits of storage. This amount of storage outperforms the combinatorial approach if \(\log |S| \cdot \frac{d+1}{d} \leq n \log n\), that is, the rate of \(S\) is at most \(\frac{d}{d+1}\). It will be shown in the sequel (Theorem 48 and Lemma 103) that sets of permutations of this rate and locality \(d\) do exist. Moreover, the use of LRC for storing the subset \(S\) with this much overhead seems to require enumerative decoding of \(S\), a procedure that eradicates the ability for quick answer to Q1 and Q2. Therefore, there exist scenarios in which the combinatorial approach outperforms the coding one, both in terms of redundancy and in terms of instant access.

In addition, simple LRCs will be used for proof of existence of optimal sets of permutations with locality. In particular, in Subsection 8.B.3.2 it will be shown that there exists a *coset* of an optimal locally recoverable code \(C\), which contains a set \(S\) of words that can be considered as permutations. However, this claim is merely existential, and does not provide any significant insights on the structure of \(S\).

---

\footnote{More precisely, the alphabet \([n]\) when seen as a subset of a large enough finite field \(\mathbb{F}_q\), over whom the construction of the LRC is possible.}
8.B.3 Bounds

Let $A(n, d)$ be the maximum size of a subset of $S_n$ with locality $d$. This section presents an upper bound and an existential lower bound on $A(n, d)$. The upper bound in Subsection 8.B.3.1 is an adaptation of a bound for LRCs ([25, Theorem 2.1], given in (8.3)). This upper bound is later improved for $d = 1$, and is attained by a certain construction in Section 8.B.4.2 to follow. To obtain an existential lower bound, in Subsection 8.B.3.2 it is shown that an optimal LRC has a coset which contains a set of permutations with the same locality as the LRC itself. For certain small values of locality, an equivalent lower bound will also be derived in Subsection 8.B.3.2 by a connection to a classical problem in chess.

Notice that in this section the combinatorial approach is considered. This clearly does not fully reflect the entire spectrum of techniques that might be used to store permutations. Nevertheless, it presents the limitation of a certain approach towards storage of permutations, which is the main one in this paper.

8.B.3.1 Upper Bounds

The bound for LRCs (8.3) can be used as-is if $n$ is a power of prime, and the set of permutations is considered as a non-linear code in $F_n^n$. By a simple adaptation of [25, Theorem 2.1] to non-linear codes, we have that a non-linear code in $F_n^n$ with locality $d$ contains at most $n^{\lfloor dn/(d+1)\rfloor}$ codewords. This bound may be improved by utilizing the combinatorial structure of permutations.

The following bound, as the one given in (8.3), assumes a non-adaptive decoder. That is, it is assumed that for a given erased entry $i$, the decoder accesses a set $I_i$ of $d$ entries, and receives all their content at once. A different approach, which is partially implemented in Section 8.B.5.2, is to access the non-erased symbols in an adaptive manner, where the locations of the latter ones depend on the content of the former ones. The following lemma, due to [25], is a variant of a classic result by [1].

Lemma 101 [25, Theorem A.1] If $G$ is a directed graph on $n$ vertices then
there exists an induced directed acyclic subgraph of $G$ on at least

$$\frac{n}{1 + \frac{1}{n} \sum_i d_{i}^{out}}$$

vertices, where $d_{i}^{out}$ is the outgoing degree of vertex $i$.

This lemma provides the following adaptation of [25, Theorem 2.1] to permutations.

**Theorem 46** $A(n, d) \leq \frac{n!}{n^{d+1}}!$.

**Proof.** Let $C \subseteq S_n$ be a set of permutations with locality $d$, and let $G$ be a directed graph whose vertex set is $[n]$, and $(i, j)$ is an edge if entry $j$ in $\pi \in C$ is required for the local correction of entry $i$. Notice that since $C$ has locality $d$, Lemma 101 implies that $G$ has an induced directed acyclic subgraph on a set $U$ of at least $\left\lceil \frac{n}{d+1} \right\rceil$ vertices. Since this subgraph is acyclic, it contains a vertex $i$ with no outgoing edges. Hence, entry $i$ is a function of entries in $[n] \setminus U$. Repeating this argument for the induced graph on $U \setminus \{i\}$, we have that there exists a vertex $i'$ with no outgoing edges to $U \setminus \{i\}$. Hence, entry $i'$ is a function of entries in $[n] \setminus (U \setminus \{i\})$. Since entry $i$ is a function of entries in $[n] \setminus U$, we have that $i'$ is also a function of entries in $[n] \setminus U$. Iterating over all vertices in $U$, we have that there are at least $\left\lceil \frac{n}{d+1} \right\rceil$ entries that depend on the other $\left\lfloor \frac{dn}{d+1} \right\rfloor$ entries.

Therefore, there exists a set of at most $\left\lfloor \frac{dn}{d+1} \right\rfloor$ entries, which determines the entire permutation. There are $\left( \frac{n}{\left\lfloor \frac{dn}{d+1} \right\rfloor} \right)!$ different ways to choose the elements in these entries, and $\left\lfloor \frac{dn}{d+1} \right\rfloor!$ ways to permute them. Therefore, the size of $C$ is at most

$$\left( \frac{n}{\left\lfloor \frac{dn}{d+1} \right\rfloor} \right)! \cdot \left\lfloor \frac{dn}{d+1} \right\rfloor! = \frac{n!}{\left\lfloor \frac{dn}{d+1} \right\rfloor! \cdot \left( n - \left\lfloor \frac{dn}{d+1} \right\rfloor \right)!} \cdot \left( \frac{dn}{d+1} \right)! \cdot \frac{n!}{\left\lfloor \frac{dn}{d+1} \right\rfloor!} = \frac{n!}{\left( n - \left\lfloor \frac{dn}{d+1} \right\rfloor \right)!} = \frac{n!}{\left\lfloor \frac{dn}{d+1} \right\rfloor!}.$$

As a simple corollary of Theorem 46, we obtain an upper bound on the rate of a set of permutations with locality $d$. Notice that by the Stirling
approximation of the factorial function, we have that \( \log(n!) \approx n \log n \), and hence the optimal rate implied by Theorem 46 is

\[
\frac{\log \left( \frac{n!}{\lceil \frac{n}{d+1} \rceil !} \right)}{\log(n!)} = 1 - \frac{\log \left( \lceil \frac{n}{d+1} \rceil ! \right)}{\log(n!)}
\]

\[
\xrightarrow{n \to \infty} 1 - \frac{n}{d+1} \cdot \log \left( \frac{n}{d+1} \right)
\]

(8.4)

The trivial subset \( C = S_n \) admits locality of \( d = n - 1 \), and attains the upper bound. In addition, the alternating group, and its complement, have locality of \( n - 2 \) (see Subsection 8.B.4.1). According to these examples, it is clear that \( A(n, n - 1) = n! \) and \( A(n, n - 2) = n!/2 \), and hence, any upper bound will coincide with the one given in Theorem 46 for \( d \in \{n - 1, n - 2\} \).

For \( d < n - 2 \) there exists a large gap between this bound and the sizes of the sets presented in this paper. This gap may be resolved for \( d = 1 \) by using a graph theoretic argument on the dependency graph in the proof of Theorem 46. In what follows, we say that a directed graph \( G \) is connected if removing the directions from the edges of \( G \) yields an undirected connected graph \( T \). If \( T \) is not connected, then every connected component of \( T \) is considered as a connected component of \( G \).

**Lemma 102** If \( G \) is a directed graph of constant out-degree one, then any connected component of \( G \) contains precisely one cycle.

**Proof.** Let \( G' \) be a connected component of \( G \). If \( G' \) contains no vertex with in-degree zero, then it is a cycle, and the claim is clear. Else, let \( v_1 \) be a node in \( G' \) with in-degree zero. Since \( G \) has constant degree one, it follows that there exists a unique path \( v_1 \to \ldots \to v_t \), such that all vertices are distinct, and \( t \) is maximal. Since \( v_t \) has out-degree one, it follows that \( G' \) contains a cycle.

If \( G' \) contains no other nodes besides \( v_1, \ldots, v_t \), then it contains precisely one cycle, and the claim follows. If \( G' \) contains no other node with in-degree zero, we have that \( G' \) contains two connected components, a contradiction. Hence, let \( v_{t+1} \) be another node in \( G \) of in-degree zero. Similarly, \( G \) also
contains a path \( v_{t+1} \to \ldots \to v_s \) with distinct nodes and a maximal \( s \) such that \( s \not\in \{v_1, \ldots, v_t\} \). The node \( v_s \) has out-degree one, and hence must be connected to another node \( u \) in \( G' \). If \( u \in \{v_{t+1}, \ldots, v_s\} \), we have that \( G' \) contains two connected components, a contradiction. Therefore, the path \( v_{t+1}, \ldots, v_s \) is connected to one of the nodes \( v_1, \ldots, v_t \), and thus no additional cycle is possible. By iterating this argument until all vertices of \( G' \) are traversed, we have that \( G' \) contains precisely one cycle.

As a result, we obtain the following bound on the maximal size of sets of permutations with locality one.

**Theorem 47** \( A(n,1) \leq n!! \equiv \prod_{i=0}^{[n/2]-1} (n-2i) \).

**Proof.** Let \( C \subseteq S_n \) be a set with locality one, and let \( G \) be a directed graph whose vertex set is the set of entries \([n]\), and \((i,j)\) is an edge if entry \( j \) in \( \pi \in C \) is required for the local correction of entry \( i \). Since \( G \) has locality one, it follow that any vertex in \( G \) has out-degree one. Furthermore, for any edge \((i,j)\), fixing value of \( \pi_j \) determines the value of \( \pi_i \).

Let \( G_1, \ldots, G_t \) be the connected components of \( G \). According to Lemma 102, each \( G_i \) contains precisely one cycle. Clearly, fixing the value of any entry in such a cycle, determines the value of all other entries in its connected component. Hence, the entire permutation \( \pi \in C \) is determined by fixing the value of \( t \) of its entries, one entry in the unique cycle in each connected component. Since each connected component contains at least two nodes, we have that the maximum size of \( C \) is

\[
n \cdot (n-2) \cdot (n-4) \cdot \ldots \cdot \left( n - 2 \left( \left\lfloor \frac{n}{2} \right\rfloor - 1 \right) \right).\]

Since the set constructed in Section 8.B.4.2 below attains the bound of Theorem 47 for \( d = 1 \), we have that \( A(n,1) = n!! \). Generalizing the techniques in the proof of Theorem 47 to any locality seems to be related to extinction problems in cellular automata on graphs [28], and might improve the bound given in Theorem 46 for many special cases.

**8.B.3.2 Lower Bound**

Optimal LRC of length \( n \) and locality \( d \) may easily be constructed over \( \mathbb{Z}_n \), the set of integers modulo \( n \). This is done by adding \( n/(d+1) \) “parity
checks” to all disjoint sets of \( d \) consecutive symbols in \( \mathbb{Z}_n^{n-n/(d+1)} \). The rate of this code attains the upper bound of \( \frac{n-n/(d+1)}{n} = \frac{d}{d+1} \), given in (8.3), and since the code is linear, all its cosets have locality \( d \) as well. Since \( n! \) of the words in \( \mathbb{Z}_n^n \) are permutations, we obtain the following existential lower bound on \( A(n, d) \).

**Theorem 48** \( A(n, d) \geq n!/n^{n/(d+1)}. \)

**Proof.** Let \( C \) be an optimal linear LRC of length \( n \) and locality \( d \) over \( \mathbb{Z}_n \). Since \( C \) is linear, it has \( \frac{n^n}{n^{d+1}} = n^{n/(d+1)} \) cosets, and each of which has locality \( d \) as well. Since \( n! \) of the words in \( \mathbb{Z}_n^n \) are permutations, it follows by the pigeonhole principle that one of the cosets of \( C \) contains a set of at least \( n!/n^{n/(d+1)} \) permutations.

The rate which is implied by Theorem 48 asymptotically attains the rate of the upper bound from Theorem 46, given in (8.4), since

\[
\frac{\log\left(\frac{n!}{n^{n/(d+1)}}\right)}{\log(n!)} = 1 - \frac{n\log n}{\log(n!)} \xrightarrow{n \to \infty} \frac{d}{d+1}.
\]

On the other hand, Theorem 48 provides a set with higher redundancy than the potential upper bound, where the redundancy of a set \( S \) is defined as \( \log(n!) - \log(|S|) \). Using the same techniques as in (8.4), we have that the redundancy of the set from Theorem 48 is

\[
\log(n!) - \log\left(\frac{n!}{n^{n/(d+1)}}\right) \approx \frac{n}{d+1} \log n,
\]

where the potential redundancy implied by Theorem 46 is

\[
\log(n!) - \log\left(\frac{n!}{\lceil n/(d+1) \rceil}\right) \approx \frac{n}{d+1} \log \frac{n}{d+1}.
\]

Therefore, it may be possible to achieve sets with redundancy up to \( n \cdot \frac{\log(d+1)}{d+1} \) bits smaller than the one obtained in Theorem 48.

**Remark 10** The proof of Theorem 48 relies on a simple construction of \( C \), an optimal LRC with low minimum Hamming distance. Similarly, it is
possible to replace $C$ with an LRC of higher minimum Hamming distance (e.g., \cite{25}) and obtain an existence proof for a set of permutations with locality and minimum Hamming distance. Since minimum distance constraints are not discussed in this paper, we choose the former approach for simplicity.

For certain small values of $d$, a similar lower bound can be derived from a well-studied problem in combinatorics, which is described in the remainder of this subsection. A Latin square of order $n$ is a square $n \times n$ matrix with entries in $\{0, \ldots , n-1\}$, such that in each row and in each column, all entries are distinct. A cyclic Latin square is a Latin square such that entry $(i, j)$ equals $(i - j) \mod n$ \cite{19}. A transversal in a Latin square is a set of positions such that no two share the same row, column, or value. A transversal in a cyclic Latin square is equivalent to the following chess problem. A semi-queen is a queen that cannot move on the north-east south-west diagonal. In an $n \times n$ toroidal chessboard, it is possible to move across the generalized diagonals $\{(i, j)|i-j \equiv t \mod n\}$ for all $t \in \{0, \ldots , n-1\}$, even if the positions are not connected in the ordinary chessboard. It is readily verified that a transversal in a cyclic Latin square of order $n$ corresponds to a configuration of $n$ non-attacking semi-queens in an $n \times n$ toroidal chess board, which in turn corresponds to a permutation $\sigma \in S_n$ such that $\text{Id} + \sigma \in S_n$, where the sum is taken mod $n$ (see Figure \ref{fig:transversal}).

For a permutation $\pi \in S_n$ let $P(\pi) \triangleq \{\sigma \in S_n|\pi + \sigma \in S_n\}$. For any two permutations $\pi_1, \pi_2 \in S_n$, by applying a proper permutation of

![Figure 8.1: The equivalence of a transversal in a cyclic Latin square, an arrangement of non attacking semi-queens in a torodial chessboard, and a permutation $\sigma$ such that Id + $\sigma$ is a permutation.](image-url)
entries we get a bijection between \( P(\pi_1) \) and \( P(\pi_2) \), and hence, \(|P(\pi_1)| = |P(\pi_2)|\). Therefore, for any \( \pi \in S_n \), the size of \( P(\pi) \) depends only on \( n \), and is thus denoted by \( t_n \). Estimating \( t_n \) is a well-known problem which was resolved only recently [13]. However, given \( \pi \in S_n \), constructing the set \( P(\pi) \) efficiently is still an open problem.

**Theorem 49** [13, Theorem 1.2] If \( n \) is an odd integer then

\[
t_n = \left( e^{-1/2} + o(1) \right) n^{1/2} / n^{n-1}.
\]

Assume that there exists an efficient algorithm \( A \) that on input \((p, i)\), where \( p \in S_n \) and \( 1 \leq i \leq t_n \), outputs the \( i \)-th permutation \( \sigma \in S_n \) such that \( p + \sigma \in S_n \). The existence of \( A \) provides an efficient algorithmic construction of an asymptotically optimal set of permutations with locality.

**Lemma 103** For an odd \( n \), an integer \( d = 2^{o(\log n)} \) such that \( d + 1 \mid n \), and integers \( i_1, \ldots, i_{d-1} \) such that \( 1 \leq i_j \leq t_n \) for all \( j \in [d-1] \), let

\[
\begin{align*}
p_1 & \in S_{\frac{n}{d+1}} \\
p_2 & \triangleq A(p_1, i_1) \\
p_3 & \triangleq A(p_1 + p_2, i_2) \\
p_4 & \triangleq A(p_1 + p_2 + p_3, i_3) \\
\vdots \\
p_d & \triangleq A \left( \sum_{j=1}^{d-1} p_j, i_{d-1} \right) \\
p_{d+1} & \triangleq \sum_{j=1}^{d} p_j,
\end{align*}
\]

and define

\[
\pi_{p_1, i_1, \ldots, i_{d-1}} \triangleq p_1 \circ \left( p_2 + \frac{n}{d+1} \right) \circ \left( p_3 + \frac{2n}{d+1} \right) \circ \left( p_4 + \frac{3n}{d+1} \right) \circ \ldots \circ \left( p_d + \frac{(d-1)n}{d+1} \right) \circ \left( p_{d+1} + \frac{dn}{d+1} \right)
\]
where $\mathbf{1}$ is the all 1’s vector of length $\frac{n}{d+1}$. The resulting set

$$S \triangleq \{\pi_{p_1,i_1,\ldots,i_{d-1}} | p_1 \in S_{n/(d+1)}, 1 \leq i_1,\ldots,i_{d-1} \leq t_n\}$$

is a set of permutations in $S_n$ with locality $d$, and optimal asymptotic rate $\frac{d}{d+1}$.

**Proof.** Since for any $k$, $2 \leq k \leq d$, we have that $p_k = A(\sum_{j=1}^{k-1} p_j, i_{k-1})$, it follows from the definition of $A$ that $p_i \in S_{\frac{n}{d+1}}$ for all $i \in \{1,\ldots,d+1\}$. Therefore, any $\pi \in S$ results from a concatenation of $d+1$ permutations on disjoint $\frac{n}{d+1}$-subsets of $\{0,\ldots,n-1\}$, and thus $S \subseteq S_n$. Since $p_{d+1} = \sum_{j=1}^{d} p_j$, it follows that any symbol of any $\pi \triangleq (\pi_0,\ldots,\pi_{n-1}) \in S$ can be computed from $d$ other symbols, since for all $j \in \{0,\ldots,\frac{n}{d+1} - 1\}$ we have

$$\pi_{\frac{dn}{d+1}+j} - \frac{dn}{d+1} = \sum_{i=1}^{d} \left(\pi_{\frac{in}{d+1}+j} - \frac{in}{d+1}\right).$$

According to Theorem 49, the size of $S$ is

$$\frac{n}{d+1}! \left(\frac{n}{d+1}\right)^{d-1} = \Theta \left(\frac{n^{12d-1}}{(\frac{n}{d+1})^{(\frac{n}{d+1}-1)(d-1)}}\right).$$

Since $d = 2^{o(\log n)}$ we have that $\frac{\log d}{\log n} \xrightarrow{n \to \infty} 0$, and thus the asymptotic rate of $S$ is

$$\frac{\log |S|}{\log(n!)} \xrightarrow{n \to \infty} \frac{(2d-1)\frac{n}{d+1} \log \frac{n}{d+1} - \left(\frac{n}{d+1} - 1\right) (d-1) \log \frac{n}{d+1}}{n \log n}$$

$$= \frac{d}{d+1} \cdot \frac{n + d - 1}{n} \cdot \frac{\log \frac{n}{d+1}}{\log n} \xrightarrow{n \to \infty} \frac{d}{d+1}.$$

A non-efficient implementation of $A$ may be obtained simply by traversing all permutations in $S_n$. However, providing an efficient implementation of $A$ requires a rigorous understanding of the structure of the permutations in $P(\text{Id})$, which seems beyond the scope of contemporary knowledge. A subset of $P(\text{Id})$ of approximate size $\sqrt{n^{\sqrt{n}}}$ is given in [10], but it is too small.

---

255
Remark 11 We note that a converse claim may also be made. That is, given an optimal set of permutations with constant locality \( d \geq 2 \), which is constructed according to the outline of Lemma 103, one may explicitly construct the set \( P(\text{Id}) \). Since an explicit construction of \( P(\text{Id}) \) is not known, this may serve as a hardness result for the construction of a set of permutations with constant locality \( d \geq 2 \). However, since the outline of the construction in Lemma 103 is highly restrictive, such a hardness result might not seem insightful enough.

8.B.4 High Rate Constructions

This section presents several simple constructions of sets of permutations with locality, some of which attain the upper bound given in Section 8.B.3.1. The well-known alternating group and its complement will be shown in Subsection 8.B.4.1 to have locality of \( n - 2 \), and attain the upper bound given in Theorem 46. Another simple set of permutations, discussed in Section 8.B.4.2, is those that may be seen as a concatenation of \( n/h \) permutations in \( S_h \), for some \( h \) which divides \( n \). The locality of the latter relies on the trivial observation that any single erasure in a permutation may be corrected without requiring additional redundancy. For \( h = 2 \), this set attains the upper bound given in Theorem 47. Subsection 8.B.4.3 shows a similar technique which achieves high locality. Subsection 8.B.4.4 and Subsection 8.B.4.5 enhance the construction of Subsection 8.B.4.2 by using Reed-Solomon codes over permutation polynomials, and by using multipermutations. The results of this section are summarized in Table 8.1 in which three locality regimes are considered for comparison.

8.B.4.1 The Alternating Group

It is widely known \([7]\) that any permutation may be represented as a product of transpositions (cycles of length two). Although many different products of transpositions may represent the same permutation, all representations of a given permutation either contain an even or an odd number of transpositions. For a permutations \( \pi \in S_n \), if the number of transpositions in any representation is even, we say that \( \pi \) is even and its sign is 1. Otherwise it
Table 8.1: Summary of the results in Section 8.B.4.

<table>
<thead>
<tr>
<th>Section</th>
<th>Technique</th>
<th>Locality</th>
<th>Asymptotic Rate</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.B.4.1</td>
<td>The alternating group.</td>
<td>$n - 2$</td>
<td>1</td>
<td>Strictly optimal.</td>
</tr>
<tr>
<td>8.B.4.2</td>
<td>Concatenation.</td>
<td>$d = O(1)$</td>
<td>$\frac{1}{d+1}$</td>
<td>Strictly optimal for $d = 1$. At least $n$ times smaller than 8.B.4.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.B.4.3</td>
<td>Range restriction.</td>
<td>$\Theta(n)$</td>
<td>$\epsilon$</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>8.B.4.4</td>
<td>Reed-Solomon codes.</td>
<td>6</td>
<td>$1/2$</td>
<td>$n = 2^k$ for some $k$.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>$1/2$</td>
<td>$n = \pm 2 \mod 5$, and a prime power.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7</td>
<td>$1/2$</td>
<td>$n = 5^k$ for some $k$.</td>
</tr>
<tr>
<td>8.B.4.5</td>
<td>Multi permutations.</td>
<td>$\Theta(1)$</td>
<td>$1/2$</td>
<td>Incomparable with 8.B.4.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Theta(n^\epsilon)$</td>
<td>$(1+\epsilon)/2$</td>
<td>Larger rate than 8.B.4.2 for the same locality.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$\Theta(n)$</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

is odd, and its sign is -1. The set of all even permutations, which forms a subgroup of $S_n$ of size $n!/2$, is called the alternating group and denoted by $A_n$. In what follows we show that the sets $A_n$ and $S_n \setminus A_n$ have locality of $n - 2$. This fact will follow from the next simple lemma.

**Lemma 104** If $\pi$ and $\sigma$ are two distinct permutations in $S_n$ whose one-liners agree on $n - 2$ entries, then one of $\{\pi, \sigma\}$ is odd and the other is even.

**Proof.** Let $\{i_j\}_{j \in [n-2]}$ be the set of entries on whom $\pi$ and $\sigma$ agree, and let $\{\alpha_j\}_{j \in [n-2]}$ be the subset of $[n]$ such that for all $j \in [n-2]$, $\pi_{i_j} = \sigma_{i_j} = \alpha_j$. Clearly, $\pi$ and $\sigma$ differ only in the arrangement of the elements in
Therefore, $\pi$ may be obtained from $\sigma$ by applying a single transposition which switches between the elements of $[n] \setminus \{\alpha_j\}_{j \in [n-2]}$, and hence $\pi$ and $\sigma$ have opposite signs.

**Corollary 28** The sets $A_n$ and $S_n \setminus A_n$ have locality of $n - 2$.

**Proof.** If a symbol of the stored permutation $\pi$ is missing, by observing any $n - 2$ of the remaining symbols there exists exactly two possibilities for $\pi$. According to Lemma 104, one of these options is an odd permutation and the other is even. Hence, restricting the system to store only permutations from either $A_n$ or $S_n \setminus A_n$, we have only one possible permutation, and thus both $A_n$ or $S_n \setminus A_n$ admit locality of $n - 2$.

Although the results in this subsection are rather simple, they shed some light on the tightness of the bound given in Theorem 46. Since $d = n - 2$ we have that $n! / [(n/d + 1)!] = n! / [(n/(n-1))!] = n!/2$, and thus $A_n$ and $S_n \setminus A_n$ are optimal sets with locality $n - 2$.

### 8.B.4.2 Concatenation of Short Permutations

 Obviously, in the one-line representation, any single symbol may easily be computed from all other symbols. This principle leads to simple sets of permutations which can be stored efficiently.

Consider the set $S$ of permutations in $S_n$ which may be viewed as a concatenation of $n/h$ shorter permutations on $h$ elements, for some integer $h$ which divides $n$. That is, their one-liner may be viewed as a concatenation of $n/h$ one-liners, each of which is a permutation of either of the sets $\{1, \ldots, h\}, \{h+1, \ldots, 2h\}$, etc. Clearly, $S$ contains $(h!)^{n/h} \cdot (n/h)!$ permutations. A subset of $S$, in which the $i$-th permutation is on the set $\{(i-1)h+1, \ldots, i \cdot h\}$, was considered in [26 Corollary 19], where it was shown to be an optimal anticode under the infinity metric $d_\infty$ (see Definition 42 in Section 8.B.5.1 to follow).

**Lemma 105** If $\pi \in S$ then any symbol $\pi_i$ can be computed from $h - 1$ other symbols, i.e., the set $S$ has locality $d = h - 1$.

**Proof.** Since $\pi \in S$ it follows that $\{\pi_{h \cdot (i/h) + 1}, \ldots, \pi_i, \ldots, \pi_{(h+1) \cdot (i/h)}\} = \{jh+1, \ldots, (j+1)h\}$. Hence, observing the value of $\pi_{h \cdot (i/h) + 1}, \ldots, \pi_{(h+1) \cdot (i/h)}$
(excluding \(\pi_i\), the range \(\{jh + 1, \ldots, (j + 1)h\}\) can be identified, and \(\pi_i\) is the missing value in it.

Note that multiple erasures can be corrected simultaneously, as long as they do not reside in the same short permutation. Two erasures from the same short permutation cannot be corrected simultaneously. In addition, Q1 can be answered trivially, and Q2 requires finding the suitable sub-permutation in \(n/h\) queries, and additional \(h\) queries to locate the desired element.

Since \(d = h - 1\), we have that \(|S| = (d + 1)!^{n/(d+1)} \cdot (n/(d + 1))!\), and for \(d = 1\) we have that

\[
|S| = 2^{n/2} \cdot (n/2)! = \left(\frac{n}{2}\right) \cdot 2 \cdot \left(\frac{n}{2} - 1\right) \cdot 2 \cdot \ldots \cdot (1) \cdot 2 = n \cdot (n - 2) \cdot (n - 4) \cdot \ldots \cdot 2 = n!!.
\]

Hence, for \(d = 1\) this construction attains the bound of Theorem 17 with equality. However, for any \(d = O(1), d \geq 2\), it can be shown that these sets do not attain the optimal rate, since they are superseded by the existential lower bound of Theorem 48.

**Lemma 106** If \(h = O(1)\), the set \(S\) (of locality \(d = h - 1\)) have rate \(\frac{1}{d+1}\) as \(n\) tends to infinity.

**Proof.**

By the construction above, we have that

\[
\frac{\log ((d + 1)!^{n/(d+1)} \cdot (\frac{n}{d+1})!)}{\log(n!)} = \frac{\log ((d + 1)!)}{d + 1} \cdot \frac{n}{\log(n)} + \frac{\log \left(\frac{n}{d+1}\right)!}{\log(n!)}.
\]

and since \(d\) is constant, it follows that

\[
\lim_{n \to \infty} \frac{\log ((d + 1)!)}{\log(n) \cdot (d + 1)} + \frac{n}{d+1} \cdot \frac{\log \left(\frac{n}{d+1}\right)}{n \log(n)} = \frac{\log ((d + 1)!)}{\log(n) \cdot (d + 1)} + \frac{1}{d+1} - \frac{\log(d + 1)}{\log(n) \cdot (d + 1)} = \frac{\log(d!)}{\log(n) \cdot (d + 1)} + \frac{1}{d+1} \to \frac{1}{d+1}.
\]
By choosing \( h = \Theta(n^\epsilon) \) for some constant \( 0 < \epsilon < 1 \), we achieve a non-vanishing rate.

**Lemma 107** If \( h = \Theta(n^\epsilon) \), the sets \( S \) (of locality \( d = h - 1 \)) have rate \( \epsilon \) as \( n \) tends to infinity.

**Proof.**

\[
\log \left( \frac{\log((d+1)! \cdot \binom{n}{d+1})}{\log(n!)} \right) n \to \infty \quad \frac{n(d+1) \log(d+1) + n \log\left(\frac{n}{d+1}\right)}{(d+1)n \log n} n \to \infty \to \epsilon.
\]

In the high locality regime, where \( d = \Theta(n) \), we may similarly prove that the rate of these codes approaches 1 as \( n \) approaches infinity.

**Lemma 108** If \( h = \Theta(n) \), the sets \( S \) (of locality \( d = h - 1 \)) have rate 1 as \( n \) tends to infinity.

### 8.B.4.3 Concatenation of Range-Restricted Permutations

In this subsection we provide a technique for producing sets of permutations with high locality \( d \geq n/2 \). For a set of symbols \( \Sigma \) let \( S(\Sigma) \) denote the set of all permutations of \( \Sigma \), that is, the set of all injective functions from \( |\Sigma| \) to \( \Sigma \). In this subsection we use the alphabet \( \Sigma = \{0, \ldots, n-1\} \), and hence \( S(\Sigma) = S_n \). Let \( h \) be an integer which divides \( n \), and for \( i \in \{0, \ldots, n/h - 1\} \) let

\[
K_i \triangleq S(\{ih, ih + 1, \ldots, (i+1)h - 1\}) \circ S([n] \setminus \{ih, ih + 1, \ldots, (i+1)h - 1\}).
\]

**Lemma 109** The set \( S^h \triangleq \bigcup_{i=0}^{n/h-1} K_i \) has locality \( d = n - h - 1 \).

**Proof.** To repair a missing symbol \( \pi_j, 0 \leq j \leq n - 1 \) in \( \pi \in S^h \), distinguish between the cases \( j \leq h - 1 \) and \( j \geq h \). If \( j \leq h - 1 \), \( \pi_j \) may clearly be computed from \( \{\pi_i\}_{i \in \{0, \ldots, h-1\} \setminus \{j\}} \). If \( j \geq h \), the set of symbols \( \{\pi_i\}_{i \in \{h, \ldots, n-1\} \setminus \{j\}} \) must contain a gap of \( h \) consecutive numbers, which are located in the prefix of \( \pi \). After identifying this gap, the missing symbol \( \pi_j \) may easily be deduced.
The set \( S^h \) contains \( \frac{n}{h} \cdot h! \cdot (n - h)! = n \cdot (h - 1)! \cdot (n - h)! \) and it does not attain the upper bound given in Theorem 46. For constant \( h \) the rate of \( S^h \) asymptotically approaches 1 as \( n \) goes to infinity, since
\[
\frac{\log(n \cdot (h - 1)! \cdot (n - h)!)}{\log(n!)} \geq \frac{\log((n - h)!)}{\log(n!)} \xrightarrow{n \to \infty} 1.
\]

Equal rate may be obtained for lower locality, where \( h = \Theta(n) \); if \( h = \delta n \) for some constant \( 0 < \delta < 1 \), then
\[
\frac{\log(n \cdot (h - 1)! \cdot (n - h)!)}{\log(n!)} \xrightarrow{n \to \infty} \delta n \log(\delta n) + (1 - \delta)n \log((1 - \delta)n) \quad n \log n
\]
\[
= \delta + (1 - \delta) = 1.
\]

An identical rate is also obtained by choosing \( h = \Theta(n^\epsilon) \). Hence, the best choice of parameters for this technique seems to be \( h = \Theta(n) \), since it results in low locality and optimal rate.

### 8.B.4.4 Extended Construction from Error-Correcting Codes

This section provides a construction of a set of permutations in \( S_n \) with locality, from two constituent ingredients. The first ingredient is a set of permutations \( S \subseteq S_{n-t} \) with locality \( d \), for some given \( t \) and \( d \). The second ingredient is an error-correcting code \( T \), in which all codewords consist of \( t \) distinct symbols.

A symbol replacement function \( f \) is an injective function which maps one alphabet to another. Given a permutation \( \pi \) and a symbol replacement function \( f \) let \( f(\pi) \) be the result of replacing the symbols of \( \pi \) according to \( f \). For a set of permutations \( S \) let \( f(S) \triangleq \{ f(\pi) | \pi \in S \} \). The construction of this section relies on the following observation.

**Observation 5** If \( S \subseteq S_{n-t} \) is a set of permutations with locality \( d \) and \( f \) is a symbol replacement function, then \( f(S) \) is a set of permutations with locality \( d \) as well.

Using a proper symbol replacement function \( f \), a permutation \( f(\pi) \) for \( \pi \in S \) is concatenated to a codeword from \( T \) to create a permutation in \( S_n \). This symbol replacement function is given in the following definition, which is followed by an example.
**Definition 39** For any integers $1 < t < n$, let $\pi$ be a permutation in $S_{n-t}$ and $e \in [n]^t$ be a word with $t$ distinct symbols $\{\sigma_1, \ldots, \sigma_t\} \triangleq E \subseteq [n]$. Let $f_E$ be the following symbol replacement function

$$f_E : [n-t] \rightarrow ([n-t] \setminus E) \cup \{n-t+1, \ldots, n-t+|E \cap [n-t]|\}$$

$$f_E(i) = \begin{cases} i, & i \notin E. \\ j, & \text{for some integer } s, i \text{ and } j \text{ are the } s\text{-smallest numbers} \\ & \text{in } E \cap [n-t] \text{ and } \{n-t+1, \ldots, n\} \setminus E, \text{ respectively.} \end{cases}$$

That is, $f_E$ maps each element which does not appear in $E$ to itself, and each element which appears in $E$ is mapped to a symbol in $\{n-t+1, \ldots, n\}$ which does not appear in $E$, in an increasing manner. Using $f_E$, define the operator $\odot$ as

$$\pi \odot e \triangleq f_E(\pi) \circ e,$$

where $\circ$ denotes the ordinary concatenation of strings.

Since symbols in $\pi$ which occur in both $\pi$ and $e$ are replaced with symbols that do not appear in either $\pi$ nor $e$, we have that $\pi \odot e$ is a permutation in $S_n$, as illustrated by the following example.

**Example 15** For $n = 7$ and $t = 3$, let $\pi = (1, 2, 3, 4)$, $e = (3, 4, 7)$, and $E = \{3, 4, 7\}$. By Definition 39 we have that

$$f_E(1) = 1, \quad f_E(2) = 2, \quad f_E(3) = 5, \quad f_E(4) = 6, \quad \text{and} \quad \pi \odot e = f_E(\pi) \circ e = (1, 2, 5, 6, 3, 4, 7) \in S_7.$$ 

The operation $\odot$ is used to extend an existing set $S \subseteq S_{n-t}$ with locality to a subset of $S_n$ with a larger locality by using an error-correcting MDS code $T$.

**Lemma 110** For integers $1 < t < n$, if $S \subseteq S_{n-t}$ is a set with locality $d$ and $T$ is an MDS code in $[n]^t$ with minimum distance $\delta$ and distinct symbols, then $S \odot T \triangleq \{s \odot e | s \in S, e \in T\} \subseteq S_n$ is a set of permutations with locality $d + t - \delta + 1$. 

---

262
Proof. Let $\pi = s \odot e$ be a permutation in $S \odot T$. To repair a missing symbol $\pi_j$ for $1 \leq j \leq n$ we distinguish between the cases $j \leq n - t$ and $j > n - t$. If $j > n - t$, by the minimum distance property of the MDS code $T$ we may obtain $\pi_j$ by accessing $t - \delta + 1$ symbols from $e$. If $j \leq n - t$, then by accessing $t - \delta + 1$ symbols from $e$ we may identify the function $f_E$ used to define the operator $\odot$ (Definition 39). Once $f_E$ is known, the symbol $\pi_j$ may be obtained by using Observation 5. 

This technique can be used to obtain explicit sets with constant locality $d \geq 2$, which are the largest ones in this paper for this locality. Unfortunately, to the best of our knowledge the asymptotic rate of these sets does not exceed $\frac{1}{2}$, and hence they are not optimal. Moreover, since a set with locality 1 also has locality $d \geq 2$ for any $d$, the sets of locality 1 from Subsection 8.B.4.2 can be used for any locality greater than 1, while obtaining rate of $\frac{1}{2}$ as well. Nevertheless, for small values of $d$ we are able to construct explicit sets with locality $d$ which contain more permutations than the sets with locality 1 from Subsection 8.B.4.2.

To provide good examples by this technique, we must construct error-correcting codes in which all codewords consist of distinct symbols. For this purpose, assume that $n$ is a power of prime, and $\{0, 1, \ldots, n - 1\}$ are representations of the elements of $\mathbb{F}_n$.

Recall that a Reed-Solomon code is given by evaluations of degree restricted polynomials on a fixed set of distinct elements from a large enough finite field. These codes contain sub-codes which are suitable for our purpose. The codewords in these sub-codes are obtained by evaluations of permutation polynomials. A permutation polynomial is a polynomial which represents an injective function from $\mathbb{F}_n$ to itself. In spite of the very limited knowledge on permutation polynomials in general, all permutation polynomials of degree at most 5 are known (see [9, Table 2]). For example, we have the following lemma.

Lemma 111 [9, Table 2]

1. If $n$ is a power of 2, then there exist at least $(n - 1)(2n + \frac{n(n^2 + 2)}{3})$ permutation polynomials of degree at most 4 over $\mathbb{F}_n$.

2. If $n$ is a prime power and $n \equiv \pm 2 \pmod{5}$, then there exist at least $n^3(n - 1)$ permutation polynomials of degree at most 5 over $\mathbb{F}_n$. 

263
3. If $n$ is a prime power and $n \equiv 0 \pmod{5}$, then there exist at least $\frac{1}{2}n^2(n-1)^2$ permutation polynomials of degree at most 5 over $\mathbb{F}_n$.

As a corollary, we obtain the following constructions.

**Example 16**  
1. Let $n$ be an integer power of 2, and let $S \subseteq S_{n-6}$ be an optimal set with locality 1 (which exists by Subsection 8.B.4.2 since $n-6$ is even). Let $T$ be a subset of a Reed-Solomon code of dimension 5 and length 6 over $\mathbb{F}_n$, which corresponds to all permutation polynomials of degree at most 4. According to Lemma 110 and Lemma 111 (part 1), the set $B_1 \triangleq S \odot T$ contains $(n-6)!! \cdot (n-1)(2n+\frac{n(n^2+2)}{4})$ permutations, and has locality 6.

2. Let $n$ be a odd prime power and $n \equiv \pm 2 \pmod{5}$ (such as 7, 17, 23, etc.), and let $S \subseteq S_{n-7}$ be an optimal set with locality 1, as above. Let $T$ be a subset of a Reed-Solomon code of dimension 6 and length 7 over $\mathbb{F}_n$, which corresponds to all permutation polynomials of degree at most 5. According to Lemma 110 and Lemma 111 (part 2), the set $B_2 \triangleq S \odot T$ contains $(n-7)!! \cdot n^3(n-1)$ permutations, and has locality 7.

3. Let $n$ be a prime power and $n \equiv 0 \pmod{5}$ (i.e., $n$ is a power of 5), and let $S \subseteq S_{n-7}$ be an optimal set with locality 1, as above. Let $T$ be a subset of a Reed-Solomon code of dimension 6 and length 7 over $\mathbb{F}_n$, which corresponds to all permutation polynomials of degree at most 5. According to Lemma 110 and Lemma 111 (part 3), the set $B_3 \triangleq S \odot T$ contains $(n-7)!! \cdot \frac{1}{2}n^2(n-1)^2$ permutations, and has locality 7.

Notice that an optimal set $A \subseteq S_n$ with locality 1, which may be seen as having any larger locality, contains $n!!$ permutations (see Section 8.B.4.2). The set $B_1$ is larger, since $n!! = (n-6)!! \cdot \Theta(n^3)$ and $|B_1| = (n-6)!! \cdot \Theta(n^4)$. Similarly, for odd values of $n$, Section 8.B.4.2 provides a set $A$ of size $(n-1)!!$ and locality 1. The sets $B_2$ and $B_3$ are larger, since $|A| = (n-1)!! = (n-7)!! \cdot \Theta(n^3)$, where both $|B_2|$ and $|B_3|$ equal $(n-7)!! \cdot \Theta(n^4)$. Hence, the construction of this section provides sets which are at least $n$ times larger than those given in Section 8.B.4.2 and have larger constant locality.
8.B.4.5 High-Locality Construction From Multi-Permutations

While constructing sets of permutations with constant locality \( d \geq 2 \) and rate above \( \frac{1}{2} \) seems hard, it is fairly easy to construct sets with such rate and locality \( d = \Theta(n^\epsilon) \), for constant \( 0 < \epsilon < 1 \). Such a set is obtained from Section 8.B.4.2 by taking \( h = \Theta(n^\epsilon) \). However, the resulting rate is

\[
\frac{\log \left( (h!)^{n/h} \left( \binom{2}{h} n! \right)^{-1/h} \right)}{\log n} \xrightarrow{n \to \infty} \frac{\log n^\epsilon}{\log n} + \frac{n^{1-\epsilon}(1 - \epsilon) \log n}{n \log n} = \epsilon,
\]

where Theorem 48 guarantees that for this locality, there exist sets with rate which tends to 1 as \( n \) tends to infinity.

In this subsection it is shown that the construction from Section 8.B.4.2 may be enhanced by using multi-permutations, achieving a rate of \( \frac{1}{2} + \frac{\epsilon}{2} \) for locality \( d = \Theta(n^\epsilon) \). The methods and notations in this subsection are strongly based on [8].

For nonnegative integers \( \ell \) and \( m \), a balanced multi-set \( \{1^m, 2^m, \ldots, \ell^m\} \) is a collection of the elements in \([\ell]\), where each element appears \( m \) times. A multi-permutation on a balanced multi-set is a string of length \( \ell m \), which is given by a function \( \sigma : [\ell m] \to [\ell] \) such that for all \( i \in [\ell] \), \( |\{j | \sigma(j) = i\}| = m \). The set of all multi-permutations is denoted by \( S_{\ell,m} \), and its size is \( \frac{(m\ell)!}{(m!)^\ell} \). To distinguish between different appearances of the same element in a multi-permutation \( \sigma \), for \( j \in [m\ell] \), \( i \in [\ell] \), and \( r \in [m] \) we denote \( \sigma(j) = i_r \), and \( \sigma^{-1}(i_r) = j \) if the \( j \)-th position of \( \sigma \) contains the \( r \)-th appearance of \( i \).

**Example 17** If \( m = 2 \) and \( \ell = 3 \) then \( \pi = (1, 1, 2, 2, 3, 3) \) is a multi-permutation on the balanced multi-set \( \{1^2, 2^2, 3^2\} \). To refer to the second appearance of 2 we say that \( \pi(5) = 2_2 \).

We are interested in multi-permutations with two appearances of each element, and therefore assume that \( m = 2 \) and \( \ell = n/2 \) (although some of the definitions in the sequel have counterparts for any \( m \) and \( \ell \) such that \( n = m\ell \)). In particular, we consider such multi-permutations in which any two appearances of the same element are not too far apart. To this end, the following definition is required.

| 265 |
Definition 40 If $\pi \in S_{n/2,2}$ and $t \in [n]$ then,

$$w(\pi) \triangleq \max_{i \in [n/2]} |\pi^{-1}(i_1) - \pi^{-1}(i_2)|,$$

and

$$B_t \triangleq \{ \pi \in S_{n/2,2} \mid w(\pi) \leq t \}.$$

That is, $w(\pi)$ indicates the maximum distance between two appearances of the same element, or alternatively, $w(\pi) - 1$ indicates the maximum number of elements between two appearances of the same element in $\pi$. For a given $t$, $B_t$ is the set of all multi-permutations in $S_{n/2,2}$ in which every two identical elements are separated by at most $t - 1$ other elements. Clearly, the multi-permutation $\pi$ which was given in Example 17 is in $B_2$.

To construct “ordinary” permutations in $S_n$ from multi-permutations in $S_{n/2,2}$ we use the term assignment of permutations. As in Subsection 8.B.4.3, for a set of elements $\Sigma$ we denote by $S(\Sigma)$ the set of all permutations of $\Sigma$ (that is, the set of all injective functions $f : \{1, \ldots, |\Sigma|\} \to \Sigma$).

Definition 41 If $\pi \in S_{n/2,2}$ and $\gamma_1, \ldots, \gamma_{n/2}$ are permutations such that $\gamma_i \in S(\{2i - 1, 2i\})$ for all $i$, then $\sigma = \pi(\gamma_1, \ldots, \gamma_{n/2})$ is the permutation in $S_n$ such that for all $1 \leq j \leq n$, if $\pi(j) = i_r$ then $\sigma(j) = \gamma_i(r)$.

Example 18 If $\pi = (1, 1, 2, 3, 2, 3) \in S_{3,2}$ and $\gamma_1 = (1, 2)$, $\gamma_2 = (4, 3)$, and $\gamma_3 = (6, 5)$ then $\sigma = \pi(\gamma_1, \gamma_2, \gamma_3) = (1, 2, 4, 6, 3, 5) \in S_6$.

Note that by choosing $h = 2$ in the construction which appears in Subsection 8.B.4.2, the resulting set $S$ can be described as

$$S = \{ \pi(\gamma_1, \ldots, \gamma_{n/2}) \mid \forall i, \gamma_i \in S(\{2i - 1, 2i\}) \text{ and } \pi \in B_1 \}.$$

Hence, the construction in the following lemma may be seen as a generalization of the construction from Subsection 8.B.4.2.

Lemma 112 For a nonnegative integer $t$, the set

$$A_t \triangleq \{ \pi(\gamma_1, \ldots, \gamma_{n/2}) \mid \forall i, \gamma_i \in S(\{2i - 1, 2i\}), \text{ and } \pi \in B_t \}$$

has locality $4t$. 

266
Proof. For $i \in [n/2]$ we say that the elements $\{2i - 1, 2i\}$ are counterparts. Assume that a symbol $\pi_i$ is erased, and let

$$D \triangleq \{\pi_{i-t}, \ldots, \pi_{i-1}, \pi_{i+1}, \ldots, \pi_{i+t}\}$$

$$\overline{D} \triangleq \{\pi_{i-2t}, \ldots, \pi_{i-1}, \pi_{i}, \pi_{i+1}, \ldots, \pi_{i+2t}\},$$

where we omit all $\pi_j$’s for which $j \notin [n]$. By the definition of $A_t$, all the counterparts of the elements in $D$ are in $\overline{D}$. However, there exists $\sigma \in D$ which is the counterpart of $\pi_i$, and hence, $\sigma$’s counterpart is not to be found in $\overline{D}\setminus\{\pi_i\}$. Therefore, computing $\pi_i$ is possible by inspecting $\overline{D}\setminus\{\pi_i\}$, and returning the counterpart of the only element in $D$ whose counterpart is not in $\overline{D}\setminus\{\pi_i\}$. The claim follows since $|\overline{D}\setminus\{\pi_i\}| = 4t$.

Using this lemma, we are able to provide a set with high locality $\Theta(n^t)$, and asymptotic rate strictly above $\frac{1}{2}$. To bound this rate from below we require a lower bound on the size of $B_t$ from Definition 40.

**Lemma 113** $|B_t| \geq \left(\frac{n}{(t/2)!}\right)^{n/t} \cdot \frac{n!}{2^{t/2}} \cdot 2^{-n/2}$.

**Proof.** Since our goal is an asymptotic computation, we may w.l.o.g. assume that $t|n$ and $2|t$. Let $E$ be the set of multi-permutations $\pi \in S_{n/2,2}$ that can be written as a concatenation $\pi = \pi^{(1)} \circ \ldots \circ \pi^{(n/t)}$ of $\frac{n}{t}$ multi-permutations on $t/2$ pairs of identical elements. A simple combinatorial calculation shows that

$$|E| \geq \prod_{i=1}^{n/t} \text{(no. of options to choose } \pi^{(i)})$$

$$\geq \left(\frac{n/2}{t/2}\right)^{n/t} \cdot \frac{t!}{2^{t/2}} \cdot \left(\frac{n/2 - t/2}{t/2}\right)^{n/t} \cdot \frac{t!}{2^{t/2}} \cdot \ldots \cdot \left(\frac{1}{t/2}\right)^{n/t}$$

$$= \left(\frac{t!}{2^{t/2}}\right)^{n/t} \cdot \frac{(n/2)!}{(n/2 - t/2)!(t/2)!} \cdot \frac{(n/2 - t/2)!}{(n/2 - 2 \cdot t/2)!(t/2)!} \cdot \ldots \cdot \frac{(t/2)!}{(t/2)!}$$

$$= \left(\frac{t!}{2^{t/2}}\right)^{n/t} \cdot \frac{(n/2)!}{(t/2)!}^{n/t} = \left(\frac{t!}{(t/2)!}\right)^{n/t} \cdot \frac{n!}{2^{t/2}} \cdot 2^{-n/2}.$$

Since clearly $E \subseteq B_t$, the claim follows.

Lemma 113 allows us to bound the asymptotic rate of the set $A_t$ which was defined in Lemma 112.

267
Theorem 50 If \( t = \Theta(n^\epsilon) \) then \[ \lim_{n \to \infty} \frac{\log |A_t|}{\log n!} \geq \frac{1}{2} + \frac{\epsilon}{2}. \]

Proof. According to Lemma 113, we have that \( |A_t| \geq |B_t| \cdot 2^{n/2} \). Using the fact that \( t = \Theta(n^\epsilon) \) and the approximation \( \log(n!) \approx n \log n \), we have

\[
\log |A_t| \geq \log(|B_t| \cdot 2^{n/2}) = \log \left( \left( \frac{t}{(t/2)!} \right)^{n/t} \cdot \frac{2^t}{2^t} \right) = \log n! - n \log t - n \frac{t}{2} \log((t/2)!)
\]

\[
= \frac{n \log(t) - n \log((t/2)!)}{t \log n!} + \frac{\log((n/2)!)}{\log n!} \rightarrow \infty \sim \frac{\epsilon \log n - \frac{\epsilon}{2} \log n + \frac{1}{2}}{\log n} + \frac{\log n - 1}{2 \log n} \rightarrow \infty \sim \frac{1}{2} + \frac{\epsilon}{2}.
\]

8.B.5 Construction of Specific Families

8.B.5.1 Light Permutations by the Infinity Norm

In this subsection it is shown that permutations which involve small-magnitude shifts (in comparison with the original file), can be stored efficiently. Answering Q1 requires downloading one symbol, whereas answering Q2 requires downloading a small number of adjacent symbols. An alternative and shorter representation, in which Q2 requires downloading one symbol and Q1 requires downloading a few, will be discussed in Subsection 8.B.6.1.

These permutations arise naturally when considering any ranking of items, in which an initial conjectured ranking is imposed, and any item is not expected to exceed its initial ranking by more than a certain bound.

Definition 42 If \( \pi \) and \( \sigma \) are two permutations in \( S_n \), then

\[ d_\infty(\pi, \sigma) \triangleq \max \{|\pi(i) - \sigma(i)|\}_{i \in [n]}. \]

If \( e \) is the identity permutation in \( S_n \) then \( \ell_\infty(\pi) \triangleq d_\infty(\pi, e) \), and for any
r ∈ {0, . . . , n − 1}, let B∞(e, r) (Br in short) be the ball of radius r around e, that is, B∞(e, r) ≜ \{π ∈ Sn | ℓ∞(π) ≤ r\}.

Recall that we denote π = (π1, . . . , πn) ≜ (π−1(1), . . . , π−1(n)), and for convenience, we say that πj = 0 for any j /∈ [n].

Providing a closed-form expression for |Br| is a notorious open problem. An efficient algorithm for computing |Br| is given in [24, Corollary 5], from which the estimation |B2| ≈ 2.16n follows [24, Example 2]. This estimation clearly implies that |Br| is (at least) exponential in n for any r. On the other hand, by [26, Theorem 18] we have that the size of an anti-code of maximum d∞-distance 2r is at most (2r + 1)n/(2r + 1). As Br is clearly an anti-code of maximum d∞-distance 2r, we have that |Br| ≤ (2r + 1)n/(2r + 1). Since the exact size of Br is not known, the rate of this set is also unknown. However, for constant r, the upper bound implies that the rate of Br goes to zero as n goes to infinity.

The locality of the set Br relies on the following series of lemmas. The first lemma shows that an erasure in π ∈ Br can be corrected by inspecting the entries in radius 2r from it.

Lemma 114 If π ∈ Br then for all j ∈ [n], πj is a function of πj−2r, . . . , πj−1, πj+1, . . . , πj+2r.

Proof. Since π ∈ Br, for any t ∈ [n] we have that t ∈ {πt−r, . . . , πt+r} and πt ∈ {t − r, . . . , t + r}. Therefore,

πj ∈ \{j − r, . . . , j, . . . , j + r\}, and
\{j − r, . . . , j, . . . , j + r\} ⊆ \{πj−2r, . . . , πj, . . . , πj+2r\}.

This implies that

\{j − r, . . . , j, . . . , j + r\} \ {πj−2r, . . . , πj−1, πj+1, . . . , πj+2r} = \{πj\},

and hence, πj may be computed given πj−2r, . . . , πj−1, πj+1, . . . , πj+2r. ■

Lemma 114 implies a correction algorithm for simultaneous erasures, in scenarios where the correction by Lemma 114 is impossible.

Lemma 115 If π ∈ Br, then any set of erasures in which any two are separated by at least 2r − 1 non-erased symbols can be corrected.
Proof. Let \( E \triangleq \{s_1, \ldots, s_t\} \) be the set of erased symbols in \( \pi \). For each \( s_i \in E \) define the radius of possibility \( R(s_i) \triangleq \{s_i - r, \ldots, s_i + r\} \) (a radius, in short), which is the set of the possible locations of the missing symbol \( s_i \). Clearly, if \( \pi \) contains a single erasure in locations \( R(s_i) \), then this erasure can be corrected to \( s_i \). Hence, it suffices to show that for every \( t \), the set \( \{R(s_i)\}_{i=1}^t \) contains at least one radius which contains a single erasure.

Assume for contradiction that every radius in \( \{R(s_i)\}_{i=1}^t \) contains at least two erasures. Notice that since the erasures are at least \( 2r - 1 \) apart, every radius contains exactly two erasures. Let \( j \) be the location of the leftmost erasure, and assume w.l.o.g that it is contained in two radii \( R(s_1) \) and \( R(s_2) \). Since both radii contain two erasures, they both must contain the erasure located to the right of \( j \), at \( j + 2r \). Hence, since the size of both radii is \( 2r + 1 \), we have that \( R(s_1) = R(s_2) = \{j, \ldots, j + 2r\} \), and thus \( s_1 = s_2 \), a contradiction. ■

Since permutations in \( B_r \) involve small-magnitude shifts, Q2 can be answered by inspecting the values of \( 2r \) locations, \( r \) to the right and \( r \) to the left of location \( i \). Permutations in \( B_r \) admit a shorter representation, for which Q2 can be answered immediately, and Q1 requires inspecting \( 2r \) other symbols (see Subsection 8.B.6.1).

8.B.5.2 MinMax Permutations

The following set of permutations, called “MinMax”, includes many natural ones. E.g., all single left (right) cyclic shifts of any prefix (suffix) of the file. These permutations arise in the context of content consumption, such as news or tweets, in which the consumer begins with an arbitrary item of a feed, and either proceeds forward or backwards from the set of consecutive items which he read so far.

**Definition 43** The set of MinMax permutations \( \mathcal{M} \) consists of two subsets denoted \( \mathcal{M}_L \) and \( \mathcal{M}_R \). The subset \( \mathcal{M}_L \) (\( \mathcal{M}_R \)) consists of all permutations in which every element is either greater than the maximal element to his left (right) by 1, or smaller than the minimal element to his left (right) by 1.
That is,

\[ \mathcal{M}_L \triangleq \left\{ \pi \in S_n \mid \forall i, \pi_i \in \{\max_{j<i} \pi_j + 1, \min_{j<i} \pi_j - 1\} \right\}, \]

\[ \mathcal{M}_R \triangleq \left\{ \pi \in S_n \mid \forall i, \pi_i \in \{\max_{j>i} \pi_j + 1, \min_{j>i} \pi_j - 1\} \right\}, \text{ and} \]

\[ \mathcal{M} \triangleq \mathcal{M}_L \cup \mathcal{M}_R. \]

**Example 19** For \( n = 7 \),

- The permutation \((3, 4, 5, 2, 1, 6, 7)\) is in \( \mathcal{M}_L \) but not in \( \mathcal{M}_R \).
- The permutation \((1, 2, 3, 4, 5, 6, 7)\) is in \( \mathcal{M}_L \) and in \( \mathcal{M}_R \).
- The permutation \((3, 5, 1, 2, 7, 6, 4)\) is neither in \( \mathcal{M}_L \) nor in \( \mathcal{M}_R \).

The following lemma provides an alternative to Definition [43] and will be used in the sequel.

**Lemma 116** \( \pi \in \mathcal{M}_L \) if and only if for all \( i \in [n] \), the set \( \{\pi_1, \ldots, \pi_i\} \) contains consecutive numbers. Similarly, \( \pi \in \mathcal{M}_R \) if and only if for all \( i \in [n] \), the set \( \{\pi_i, \ldots, \pi_n\} \) contains consecutive numbers.

**Proof.** For \( \pi \in \mathcal{M}_L \) we prove by induction on \( i \) that the set \( \{\pi_1, \ldots, \pi_i\} \) contains consecutive numbers. For \( i = 1 \) the claim is clear. For an arbitrary \( i > 1 \), by the induction hypothesis we have that \( \{\pi_1, \ldots, \pi_{i-1}\} \) is a set of consecutive numbers. By the definition of \( \mathcal{M}_L \) we have that \( \pi_i \in \{\max_{j<i} \pi_j + 1, \min_{j<i} \pi_j - 1\} \), and hence the set \( \{\pi_1, \ldots, \pi_i\} \) is a set of consecutive numbers as well.

On the other hand, if for all \( i \) the set \( \{\pi_1, \ldots, \pi_i\} \) contains consecutive numbers, we show by induction on \( i \) that \( \pi_i \in \{\max_{j<i} \pi_j + 1, \min_{j<i} \pi_j - 1\} \).

For \( i = 1 \) the claim is clear. For an arbitrary \( i > 1 \), by the induction hypothesis we have that \( \pi_{i-1} \in \{\max_{j<i-1} \pi_j + 1, \min_{j<i-1} \pi_j - 1\} \). W.l.o.g assume that \( \pi_{i-1} = \max_{j<i-1} \pi_j + 1 \), and hence \( \pi_{i-1} > \pi_j \) for all \( j < i - 1 \). Therefore, since the set \( \{\pi_1, \ldots, \pi_i\} \) contains consecutive numbers, we have that either \( \pi_i > \pi_{i-1} \), and hence \( \pi_i = \max_{j<i} \pi_j + 1 \), or \( \pi_i < \pi_{i-1} \), and hence \( \pi_i = \min_{j<i} \pi_j - 1 \). For the case \( \pi_{i-1} = \min_{j<i-1} \pi_j - 1 \), the proof is similar, and if \( \pi \in \mathcal{M}_R \), the proof is symmetric.
Corollary 29 If \( \pi \in M_L \) then \( \pi_n \in \{1, n\} \), and if \( \pi \in M_R \) then \( \pi_1 \in \{1, n\} \).

Proof. If \( \pi \in M_L \), by Lemma 116 we have that \( \{\pi_1, \ldots, \pi_{n-1}\} \) is a set of consecutive numbers from \([n]\), and hence it is either \( \{1, \ldots, n-1\} \) or \( \{2, \ldots, n\} \). Therefore, \( \pi_n \in \{1, n\} \). If \( \pi \in M_R \), the proof is symmetric. ■

Albeit the simple structure of their elements, the sets \( M_L \) and \( M_R \) are rather large, as shown below.

Lemma 117 \( |M_L| = |M_R| = 2^{n-1} \).

Proof. Let \( M_{L,i} \) be the set \( M_L \) for \( n = i \), and let \( M_{L,i} \) be its size. According to Corollary 29, for any \( i \), if \( \pi \in M_{L,i} \) then \( \pi_i \in \{1, i\} \). Therefore, \( \pi \in M_{L,i} \) corresponds to exactly two permutations \( \pi^{(1)}, \pi^{(2)} \in M_{L,i+1} \). The first permutation is \( \pi^{(1)} = (\pi_1, \ldots, \pi_i, i+1) \), where the second is \( \pi^{(2)} = (\pi_1+1, \ldots, \pi_i+1, 1) \). Since this mapping clearly covers the entire set \( M_{L,i+1} \), we have that \( M_{L,i+1} = 2M_{L,i} \), and since \( M_{L,1} = 1 \), the claim follows. The proof for \( M_R \) is symmetric. ■

Lemma 118 \( M_L \cap M_R = \{(1, \ldots, n), (n, \ldots, 1)\} \).

Proof. If \( \pi \in M_L \cap M_R \), then according to Corollary 29 we have that \( \{\pi_1, \pi_n\} = \{1, n\} \). By Definition 43, if \( (\pi_1, \pi_n) = (1, n) \) then \( \pi = (1, \ldots, n) \), and if \( (\pi_1, \pi_n) = (n, 1) \) then \( \pi = (n, \ldots, 1) \). ■

Corollary 30 \( |M| = 2^n - 2 \).

Proof. By Lemma 117 and Lemma 118 \( |M| = |M_L| + |M_R| - |M_L \cap M_R| = 2^n - 2 \).

The locality of \( M_L \) and \( M_R \) relies on the following lemma.

Lemma 119 If \( \pi \in M_L \), then every \( \pi_i \) is a function of at most three other symbols of \( \pi \).

Proof. We distinguish among the following three cases.

\( i = 1 \). We compute \( \pi_1 \) from \( \pi_2 \) and \( \pi_3 \). Notice that by Definition 43 \( |\pi_3 - \pi_2| \leq 2 \), since otherwise \( \pi_3 \) does not comply with the definition.

Given the value of \( \pi_2 \), we have that either \( \pi_1 = \pi_2 + 1 \) or \( \pi_1 = \pi_2 - 1 \).
If \( \pi_3 = \pi_2 \pm 1 \), we are done. If \( \pi_3 - \pi_2 = 2 \), then by Definition 43 we have that \( \pi_3 = \max\{\pi_1, \pi_2\} + 1 \), and hence \( \pi_1 = \pi_2 + 1 \). Similarly, if \( \pi_2 - \pi_3 = 2 \) we have that \( \pi_3 = \min\{\pi_1, \pi_2\} - 1 \), and hence \( \pi_1 = \pi_2 - 1 \).

**i = n.** We compute \( \pi_n \) from \( \pi_{n-1} \) and \( \pi_{n-2} \). By Corollary 29 we have that \( \pi_n \in \{1, n\} \), and hence, if \( \{1, n\} \cap \{\pi_{n-1}, \pi_{n-2}\} \neq \emptyset \), we are done. Else, if \( \pi_{n-1} > \pi_{n-2} \), we have that \( \pi_{n-1} > \pi_j \) for all \( j < n - 1 \). Since \( \pi_{n-1} \) is larger than \( n - 2 \) numbers in \( [n] \) we have that \( \pi_{n-1} \in \{n-1, n\} \). Since the case \( \pi_{n-1} = n \) was already considered, we have that \( \pi_{n-1} = n - 1 \), and hence, \( \pi_n = n \). On the other hand, if \( \pi_{n-1} < \pi_{n-2} \), then similarly, \( \pi_{n-1} < \pi_j \) for all \( j < n - 1 \), and hence \( \pi_n = 1 \).

**i \notin \{1, n\}.** We compute \( \pi_i \) from \( \pi_{i-1}, \pi_{i+1} \), and if \( i > 2 \) we require an arbitrary \( \pi_j \) for \( j < i - 1 \). Let \( A \triangleq \{\pi_1, \ldots, \pi_i\} \) and \( n \triangleq \{\pi_1, \ldots, \pi_{i-2}\} \). According to Lemma 116 and Definition 43 if \( \pi_{i-1} < \pi_{i+1} \) we have that \( A = \{\pi_{i+1} - i, \ldots, \pi_{i+1} - 1\} \), and if \( \pi_{i-1} > \pi_{i+1} \), we have that \( A = \{\pi_{i+1} + 1, \ldots, \pi_{i+1} + i\} \). If \( i = 2 \), and thus \( n = \emptyset \), we have that \( \pi_i \) is the only element in the singleton \( A \setminus \{\pi_{i-1}\} \). Otherwise, we examine any \( \pi_j \) for \( j < i - 1 \) in order to distinguish between the cases \( n = \{\pi_{i-1} + 1, \ldots, \pi_{i-1} + i - 2\} \) and \( n = \{\pi_{i-1} - (i - 2), \ldots, \pi_{i-1} - 1\} \). Knowing \( n \), we have that \( \pi_i \) is the only symbol in the singleton \( A \setminus (n \cup \{\pi_{i-1}\}) \).

Since permutations in \( M_R \) are entirely symmetric to the ones in \( M_L \), we have that by replacing \( i \) for \( n - i + 1 \) in the proof of Lemma 119 we are able to prove the following.

**Lemma 120** If \( \pi \in M_R \), then every \( \pi_i \) is a function of at most three other symbols of \( \pi \).

Notice that the repair algorithms, which are induced by Lemma 119 slightly differ for \( \pi \in M_L \) and \( \pi \in M_R \). Therefore, the user must know if the stored permutation belongs to \( M_L \) or to \( M_R \). Clearly, this information may be stored in the system with one additional bit. Alternatively, it can also be differed by querying either one of \( \pi_1, \pi_n \), as shown in the following lemma.
Lemma 121 For $\pi \in \mathcal{M}$, if $\pi_1 \notin \{1, n\}$ then $\pi \in \mathcal{M}_L$, and if $\pi_1 \in \{1, n\}$ then $\pi \in \mathcal{M}_R$. Similarly, if $\pi_n \notin \{1, n\}$ then $\pi \in \mathcal{M}_R$, and if $\pi_n \in \{1, n\}$ then $\pi \in \mathcal{M}_L$.

Proof. If $\pi_1 \notin \{1, n\}$ then by Corollary 29 we have that $\pi \notin \mathcal{M}_R$, and hence $\pi \in \mathcal{M}_L$. On the other hand, if $\pi_1 \in \{1, n\}$ then $\pi$ may be either in $\mathcal{M}_L$ or in $\mathcal{M}_R$. However, if $\pi \in \mathcal{M}_L$ and $\pi_1 \in \{1, n\}$, it follows by Definition 43 that either $\pi = (1, \ldots, n)$ or $\pi = (n, \ldots, 1)$, which by Lemma 118 implies that $\pi \in \mathcal{M}_R$. The second part of the proof is symmetric.

As a corollary of Lemma 119, Lemma 120, and Lemma 121 we have the following.

Corollary 31 If $\pi \in \mathcal{M}$, then every $\pi_i$ is a function of at most four other symbols of $\pi$.

In the sequel we analyse patterns of erasures that can be corrected simultaneously. To this end, we devise the following graph $G$ (Figure 8.2), which follows from Lemma 119.

The vertices of $G$ correspond to subsets of locations in the permutation, where we denote $i$ instead of $\{i\}$ for convenience. A directed edge $(i, S)$ exists if the repair of $\pi_i$ requires precisely one (arbitrary) symbol $\pi_j$, $j \in S$. An analogue graph may be achieved for $\pi \in \mathcal{M}_R$ by replacing $i$ for $n - i + 1$ in each set.

![Figure 8.2: The dependency graph between the symbols of a MinMax permutation $\pi \in \mathcal{M}_L$. Nodes represent subsets of symbol locations, where $\{i\}$ is denoted by $i$. An edge $(i, S)$ indicates that the repair of symbol $\pi_i$ requires exactly one arbitrary symbol $\pi_j$, $j \in S.$](image-url)
For a vertex \( v \) in \( G \) which corresponds to a singleton, let \( \Gamma(v) \) be its set of outgoing neighbors. A vertex \( u \) which corresponds to a subset \( S \) is called active, if there exists \( j \in S \) such that \( \pi_j \) is available to the user. According to Lemma \ref{lem:repairability}, a symbol \( \pi_i \) is repairable if all nodes in \( \Gamma(i) \) are active. The following lemma presents the sets of erasures that may be simultaneously corrected. A symmetric analogue of Lemma \ref{lem:correctness} for \( \pi \in \mathcal{M}_R \) may be proved similarly.

**Lemma 122** Given a set of erased symbols, a permutation \( \pi \in \mathcal{M}_L \) can be reconstructed if every path of erased symbols in \( G \) is finite, and terminates with a node \( i \) such that every node in \( \Gamma(i) \) is active.

**Proof.** Assume that \( \{\pi_i\}_{i \in E} \) is a set of missing symbols, such that every path in \( G \) whose vertex set is contained in \( E \), terminates with a node \( i \) such that \( \Gamma(i) \) are active. Using induction on the maximum length of those paths, we get that \( \pi \) can be reconstructed. If this maximum length is 1, \( \pi \) clearly can be reconstructed. If it is greater than 1, we may repair all terminal nodes in all maximum length paths, and use the induction hypothesis. \( \blacksquare \)

### 8.B.6 Efficient Representation of Permutations

#### 8.B.6.1 Light Permutations by the Infinity Norm - An Alternative Representation

A permutation \( \pi \in \mathcal{B}_r \) can be mapped to a list of size \( n \), which indicates the displacement \( \pi(i) - i \) for each \( i \in [n] \). This mapping, denoted by \( s(\pi) \), requires \( n(\log r + 1) \) bits. By storing this representation rather than the one-liner, we may answer Q2 simply by inspecting \( s(\pi)_i \) and returning \( i + s(\pi)_i \). Since permutations in \( \mathcal{B}_r \) involve small-magnitude shifts, to answer Q1 we may look at \( s(\pi)_j \) for \( j \in \{i - r, \ldots, i + r - 1\} \), return the unique \( j \) such that \( j + s(\pi)_j = i \), or return \( i + r \) if no such \( j \) exists. The equivalent claims of Lemma \ref{lem:infinite} and Lemma \ref{lem:finite} are given below.

**Lemma 123** If \( \pi \in \mathcal{B}_r \) then for all \( j \in [n] \), \( s(\pi)_j \) is a function of \( s(\pi)_{j-2r}, \ldots, s(\pi)_{j-1}, s(\pi)_{j+1}, \ldots, s(\pi)_{j+2r} \).

**Proof.** The missing entry \( s(\pi)_j \), specifying the shift of the element \( j \), is computed by maintaining a binary array \( A \) which indicates which are the...
known “occupied” positions in the one-liner of $\pi$. Once we are left with a single non-occupied position $A_\ell$ for some $\ell$, we infer that element $j$ could only be located at $\ell$, and compute its shift.

Formally, initialize an array $A$ of $2r + 1$ zeros, whose entries are indexed by $\{j - r, \ldots, j, \ldots, j + r\}$, which are all possible positions of $j$. Set entry $A_i$ to one if and only if there exists $t \in \{j - 2r, \ldots, j - 1, j + 1, \ldots, j + 2r\}$ such that $t + s(\pi)_t = i$. Consequently, entry $A_i$ will indicate if one of the elements $\{j - 2r, \ldots, j - 1, j + 1, j + 2r\}$, whose location is known from $s(\pi)_{j - 2r}, s(\pi)_{j - 1}, s(\pi)_{j + 1}, \ldots, s(\pi)_{j + 2r}$, is located in position $i$.

Since $\pi \in B_r$, for each $i \in \{j - r, \ldots, j, \ldots, j + r\}$ there exist a unique $t \in \{j - 2r, \ldots, j, \ldots, j + 2r\}$ such that $t + s(\pi)_t = i$. Therefore, there exists a unique entry $\ell$ in $A$ which remains zero, and hence, $s(\pi)_j = \ell - j$. □

**Lemma 124** If $\pi \in B_r$, then any set of erasures in $s(\pi)$, in which any two are separated by at least $2r - 1$ non-erased symbols, can be corrected.

**Proof.** As in the proof of Lemma 123 we use an auxiliary array whose entries specify which are the occupied positions in the one-liner of $\pi$.

Initialize an array $A$ of $n$ zeros, and for each non-erased symbol $s(\pi)_j$ set $A_{s(\pi)_j + j}$ to one. Let $E = \{u_i\}_{i=1}^t$ be the set of indices of zeros in $A$, that is, $A_\ell = 0$ if and only if $\ell \in E$, and notice that the number of zeros in $A$ equals the number of erasures. For each $i \in [t]$ define the radius of possibility $R(i) \triangleq \{\max\{u_i - \ell, 1\}\}_{\ell=r}^{\ell=r}$ (radius, in short). Clearly, if there exists a single erasure among $\{s(\pi)_\ell \mid \ell \in R(i)\}$, say in $s(\pi)_{\ell_0}$, then it may be corrected to $u_i - \ell_0$. Hence, it suffices to show that for any $t$, there exists an $i \in [t]$ such that there is a single erasure among $\{s(\pi)_\ell \mid \ell \in R(i)\}$.

Assume for contradiction that all radii contain at least two erasures. Since any two erasures have at least $2r - 1$ non-erased symbol between them, and since all radii consist of $2r + 1$ consecutive integers, we have that each radius contains exactly two erasures. Now let $e_1 < \ldots < e_t$ be the indices of the erasures in $s(\pi)$, and notice that each radius must contain exactly $e_i$ and $e_{i+1}$ for some $i \in [t - 1]$. Therefore, the maximum number of radii is $t - 1$, a contradiction. □

**8.B.6.2 Supporting Arbitrary Powers - a framework**

In this section we present a framework for storing a permutation $\pi$ from a set $T$, while allowing the user to query $\pi^k(i)$ for any $i \in [n]$ and any integer
This framework is strongly based on [20], which provides an algorithm for representing any permutation $\pi \in S_n$. In [20], the queries $\pi^k(i) =$? are answered using a rank-select data structure which is maintained separately from $\pi$. This data structure requires as much as $n + o(n)$ additional bits of storage, and is required in its entirety for any operation. For this reason, we modify the techniques of [20] to achieve a scheme which is applicable for distributed storage systems. In addition, to obtain locality, this scheme can be combined with the techniques that were developed in earlier sections.

A permutation $\pi \in S_n$ may be given in its disjoint cycle representation. This representation is not unique, as each cycle may be cyclically shifted, and cycles may be permuted. For a disjoint cycle representation $y(\pi)$ of a permutation $\pi$, let $\overline{y(\pi)}$ denote the string obtained by omitting all brackets from $y(\pi)$, e.g.,

$$y(\pi) = (123)(45)$$

$$\overline{y(\pi)} = 1 2 3 4 5.$$

Clearly, $\overline{y(\pi)}$ may be considered as a one-liner of a different permutation. Relying on this principle, we present a framework for storing permutations.

Let $S \subseteq S_n$ be a set of permutations with locality $d$, enabling an answer for Q1 by downloading $q_1$ symbols, and an answer for Q2 by downloading $q_2$ symbols. In addition, let $t$ be the maximum length of a cycle in a permutation in $S$. Let $\gamma(S)$ be the set of all permutations $\pi \in S_n$ that have a disjoint cycle representation $y(\pi)$, such that the one-liner $\overline{y(\pi)}$ is in $S$. That is,

$$\gamma(S) \triangleq \left\{ \pi \in S_n \mid \exists \overline{y(\pi)} \text{ s.t. } \overline{y(\pi)} \in S \right\}.$$

Notice that $S$ and $\gamma(S)$ are not equal. E.g., for $n = 4$, by Corollary 30 there are 14 MinMax permutations, but a simple computer search shows that 23 permutations $\pi \in S_4$ have a disjoint cycle representation $y(\pi)$ such that $\overline{y(\pi)}$ is a one-liner of a MinMax permutation.
We encode a permutation $\pi \in \gamma(S)$ to a sequence of $n$ triples $\{(\psi_i, c_i, \ell_i)\}_{i=1}^n$, where for all $i$, $\psi_i \in [n]$ and $c_i, \ell_i \in [t]$. Each triple is then placed in a storage node. Let $i_1, i_2, \ldots$ be integers such that the representation $y(\pi) = (i_1, \pi(i_1), \pi^2(i_1), \ldots)(i_2, \pi(i_2), \pi^2(i_2), \ldots)$ satisfies $y(\pi) \in S$, and let $\psi \in S$ be the permutation whose one-liner is $\overline{y(\pi)}$. That is, $\psi = (\psi_1, \ldots, \psi_n) \triangleq (i_1, \pi(i_1), \pi^2(i_1), \ldots, i_2, \pi(i_2), \pi^2(i_2), \ldots)$.

For a symbol $\psi_i$, let $c_i$ be the length of the cycle in $y(\pi)$ that contains $\psi_i$, and let $\ell_i$ be the location of $\psi_i$ in this cycle.

If node $v_i$ fails, use the repair algorithm for $S$ to obtain $\psi_i$. In addition, download $c_{i-1}, \ell_{i-1}, c_{i+1}, \ell_{i+1}$ from nodes $v_{i-1}, v_{i+1}$ and obtain $c_i, \ell_i$ by the following guidelines.

if $c_{i-1} = \ell_{i-1}$ then $
\begin{align*}
c_i &= \ell_i = 1 & &\text{if } \ell_{i+1} = 1, \\
c_i &= c_{i+1}, \ell_i = 1 & &\text{if } \ell_{i+1} \neq 1,
\end{align*}$

if $c_{i-1} \neq \ell_{i-1}$ then $c_i = c_{i-1}, \ell_i = \ell_{i-1} + 1$.

Lemma 125 This system is capable of providing $\pi^k(i)$ for any $\pi \in S_n$, $i \in [n]$, and any integer $k$ by downloading $(q_1 + q_2) \log n + 2 \log t$ bits.

Proof. Given $i$ and $k$, perform the following algorithm.

1. Find $j \triangleq \psi(i)$ by downloading $q_1 \log n$ bits.
2. Download $c_j$ and $\ell_j$ from $v_j$ ($2 \log t$ bits).
3. Compute $s \triangleq j - \ell_j + (\ell_j + k) \mod c_j$.
4. Return $\psi_s = \psi^{-1}(s)$ by downloading $q_2 \log n$ bits.

Clearly, these steps require downloading $(q_1 + q_2) \log n + 2 \log t$ bits. We are left to show that $\psi_s = \pi^k(i)$. Step 1 of the algorithm provides the location of symbol $i$ in the one-liner $\psi$, since $\psi_j = \psi^{-1}(j) = i$. In step 3, the expression $j - \ell_j$ is the starting point of the cycle which contains symbol $i$, and by adding $(\ell_j + k) \mod c_j$ we have the location $s$ of the element $\pi^k(i)$.
Example 20 Let $y(\pi) = (1 \ 0 \ 2)(4 \ 3 \ 5)(7 \ 6)$ be a disjoint cycle representation of a permutation $\pi$ on eight elements. Clearly, $y(\pi) = (1 \ 0 \ 2 \ 4 \ 3 \ 5 \ 7 \ 6)$ is a one-liner of a permutation $\psi \in B_\infty(e, 1) = B_1$ (see Section 8.B.5.1), and thus $q_1 = 1$ and $q_2 = 2$. The permutation $\pi$ is stored on $v_0, \ldots, v_7$ as follows.

<table>
<thead>
<tr>
<th>$v_i$</th>
<th>$v_1$</th>
<th>$v_2$</th>
<th>$v_3$</th>
<th>$v_4$</th>
<th>$v_5$</th>
<th>$v_6$</th>
<th>$v_7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi_i$</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>7</td>
</tr>
<tr>
<td>$c_i$</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$\ell_i$</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
</tbody>
</table>

For example, follow the algorithm in Lemma 125 for answering $\pi^2(3)$ (that is, $i = 3$ and $k = 2$). Using the algorithm for $B_1$, in step 1 the user finds the location of 3 in the one-liner, which is $j = 4$. The user later downloads $(c_4, \ell_4) = (3, 1)$ from $v_4$, computes $s = 4 - 1 + (1 + 2) \mod 3 = 3$. In step 4 the user downloads $\psi_3 = 4$ from $v_3$, which equals $\pi^2(3)$.

The obvious drawback of this system is the large storage overhead of $2n \log t$, on top of the $n \log n$ bits which are required to store $\psi$. With no known restriction on $t$, the total storage in this system is approximately $3n \log n$. However, a considerable advantage is the ability to obtain any power of the stored permutation, with a small computational overhead.

For a given set $S \subseteq S_n$, natural questions to ask are what is the size of $\gamma(S)$, and what is the structure of the permutations in $\gamma(S)$. The answers to these two questions seem rather involved. Therefore, in Table 8.2 we provide an insight towards the answer to the former, using a computer search, and for small values of $n$. It is evident from this table that the size of $\gamma(S)$ is most likely much larger than the size of $S$.

8.B.6.3 Beating LRCs - a proof of concept

Consider the scenario in which we would like to store any permutation using minimal redundancy. A possible simple solution is to use LRCs (Subsection 8.B.2), which will require at least $\frac{n}{d}$ additional symbols, or $\frac{n}{d} \log n$ additional bits, according to (8.3). This minimal redundancy may be achieved by adding a “parity check” symbol for any of $n/d$ disjoint $d$-subsets of symbols. In this section, it is shown that the fact that we are operating over
permutations may be utilized to reduce the redundancy for $d \in \{2, 3\}$. Although the improvement is highly negligible, it may be shown to achieve the information theoretic lower bound, and may constitute a proof of concept for future research.

For locality $d = 2$, we present the following storage scheme, where we assume for simplicity that $n$ is even. To store a permutation $\pi \in S_n$, encode it as follows

$$(\pi_1, \ldots, \pi_n) \mapsto (\sigma_1, \ldots, \sigma_{n+n/2}) \triangleq (\pi_1, \ldots, \pi_n, \pi_1 - \pi_2, \pi_3 - \pi_4, \ldots).$$

Notice that since $\pi$ is a permutation, each of the elements $\sigma_{n+1}, \ldots, \sigma_{n+n/2}$ may be represented using $\log(n-1)$ bits. For each $i \in \{1, \ldots, \lceil n/2 \rceil\}$, any one erasure from $\{\sigma_{2i}, \sigma_{2i-1}, \sigma_{n+i}\}$ can be repaired using the other two non-erased symbols.

For locality $d = 3$, we assume that $n$ is a power of prime and the entries $\{0, 1, \ldots, n-1\}$ of the permutation are the elements of the finite field $\mathbb{F}_n$. We use the function $f(x, y, z) = \frac{x-y}{z-y}$, and the following lemma.

**Lemma 126** For field elements $a_1, a_2, a_3$, $f(a_1, a_2, a_3) \in \{2, \ldots, n-1\}$ if and only if $a_1, a_2, a_3$ are distinct.

**Proof.** If $f(a_1, a_2, a_3) \in \{2, \ldots, n-1\}$ then $f$ is well-defined over $(a_1, a_2, a_3)$, and hence $a_2 \neq a_3$. In addition, if $a_1 = a_2$ then $f(a_1, a_2, a_3) = 0$, and similarly, if $a_1 = a_3$ then $f(a_1, a_2, a_3) = 1$. Hence, $a_1, a_2, a_3$ are distinct.

| $n$ | $n!$ | $|\mathcal{M}|$ (Definition 43) | $|\gamma(\mathcal{M})|$ | $|\mathcal{B}_1|$ (Definition 42) | $|\gamma(\mathcal{B}_1)|$ | $|\mathcal{B}_2|$ | $|\gamma(\mathcal{B}_2)|$ |
|-----|-----|-------------------------------|-----------------|-----------------|-----------------|--------|-----------------|
| 3   | 6   | 6                            | 6               | 3               | 6               | 6      | 6               |
| 4   | 24  | 14                           | 23              | 5               | 22              | 14     | 24              |
| 5   | 120 | 30                           | 99              | 8               | 66              | 31     | 117             |
| 6   | 720 | 62                           | 400             | 13              | 192             | 73     | 567             |
| 7   | 5040| 126                          | 1532            | 21              | 560             | 172    | 2371            |
| 8   | 40320| 254                         | 5713            | 34              | 1660            | 400    | 9262            |

Table 8.2: Sizes of certain permutations sets for small values of $n$. 

Technion - Computer Science Department - Ph.D. Thesis PHD-2017-01 - 2017
Conversely, if \(a_1, a_2, a_3\) are distinct, then \(f(a_1, a_2, a_3)\) is well-defined, and cannot output neither 0 nor 1.

According to Lemma 126, when \(f\) is applied over a subset of symbols of a permutation, the result may be represented by \(\log(n - 2)\) bits. This gives rise to the following scheme. For simplicity assume that \(3|n\), although this scheme may be slightly changed to fit any prime power. To store \(\pi \in S_n\), encode it as follows

\[
(\pi_1, \ldots, \pi_n) \mapsto (\sigma_1, \ldots, \sigma_{n+n/3}) \triangleq (\pi_1, \ldots, \pi_n, f(\pi_1, \pi_2, \pi_3), \ldots, f(\pi_{n-2}, \pi_{n-1}, \pi_n)).
\]

Clearly, for each \(i \in \{1, \ldots, n/3\}\), any one erasure from \(\{\sigma_{3i}, \sigma_{3i-1}, \sigma_{3i-2}, \sigma_{n+i}\}\) can be repaired using the other three non-erased symbols.

The above schemes use \(n \log n + \frac{n}{3} \log(n - d + 1)\) bits to store the \(n \log n\) bits of any permutation \(\pi\), for \(d \in \{2, 3\}\). For comparison, using the corresponding LRC requires \(n \log n + \frac{n}{3} \log n\) bits.

Clearly, over an alphabet of size \(n\), the information theoretic bound on the amount of required redundancy for a single-erasure correcting code is \(\log n\). Note that with the additional assumption that the string contains \(d\) distinct symbols, this bound reduces to \(\log(n - d + 1)\). The latter bound is achieved by the above schemes for \(d \in \{2, 3\}\). We conjecture that for any \(d\), there exists a proper redundancy function \(f\) which allows single-erasure correction. That is, the function \(f\) provides output from a set of size \(n - d + 1\), when applied over inputs that contain distinct values.

### 8.B.7 Discussion and Open Problems

In this paper we discussed locality in permutations, motivated by applications in distributed storage and rank modulation coding. We have discussed two aspects of this problem, the combinatorial one and the coding one. In the combinatorial aspect, we discussed locality in permutations without any encoding, and in the coding aspect we allowed the permutation to be encoded in order to obtain this locality.

In the combinatorial aspect, we provided upper and lower bound for the maximal size of a set of permutations with locality, provided several simple constructions with high rate, and several interesting constructions with low rate. In the coding aspect we presented a method of encoding
certain permutations in order to obtain locality, and to support arbitrary powers of the permutation. We have concluded with a proof of concept which shows that any permutation may be stored with smaller redundancy than an ordinary string.

Throughout the paper we assumed that low query complexity is to be maintained. Clearly, if no such constraint is assumed, any permutation can be represented using $\lceil \log(n!) \rceil$ bits, and stored using an LRC. However, when a query complexity requirement is imposed, there seems to be much more to be studied, and our results are hardly adequate comparing with the potential possibilities.

For simplicity, we assumed that each node stores a single symbol from $[n]$, and focused on symbol locality. This convention may be adjusted to achieve storage systems with different parameters. E.g., in Subsection 8.B.1.2 we considered permutation that can be considered as a concatenation of permutations on $h$ elements. These permutations may alternatively be stored on $h$ nodes, where node $i$ stores all $i$-th elements of the concatenated permutations. In this system any single node failure may be corrected by downloading the content of all other nodes. To achieve better locality, the data may be partitioned among a larger number of nodes. Techniques such as this open a gateway towards the equivalent of array codes for permutations, which constitutes a vast area for future consideration as well.

Finally, we list herein a few specific open problems which were left unanswered in this paper.

1. Close the gap between the upper bound in Theorem 46 and the lower bound in Theorem 48, potentially by using the methods of Theorem 47.

2. Provide an explicit construction of sets with constant locality $d \geq 2$ and optimal rate $\frac{d}{d+1}$. The existence of these sets is guaranteed by Theorem 48.

3. Find the connection between a set of permutations $S$ and its corresponding $\gamma(S)$ (Section 8.B.6.2).

4. Find a proper “parity” function $f$ for any $d$ in Section 8.B.6.3.

5. Find additional large sets of permutations that have good locality.
6. Explore the locality of permutations under different representation techniques.

7. Endow $S_n$ with a suitable metric, and explore the locality of codes with a good minimum distance by this metric.
Bibliography


Part III

Unpublished Papers
Chapter 9

Asymptotically Optimal Regenerating Codes Over Any Field

Netanel Raviv

9.A ArXiV Version

Abstract

Optimal regenerating codes exist for all possible system parameters. However, known constructions require large field size, and hence may be hard to implement in practice. By using notions from the theory of extension fields, we obtain two explicit constructions of regenerating codes. These codes approach the cut-set bound as the reconstruction degree increases, and may be realized over any given field if the file size is large enough. Since distributed storage systems are the main purpose of regenerating codes, this file size restriction is trivially satisfied in most conceivable scenarios. The first construction attains the cut-set bound at the MBR point asymptotically for all parameters, whereas the second one attains the cut-set bound at the MSR point asymptotically for low-rate parameters. 

---

1This work was done while Netanel Raviv was a visiting student at the University of Toronto, under the supervision of Prof. Frank Kschischang. It is a part of his Ph.D.
9.A.1 Introduction

Since the emergence of cloud storage platforms, distributed storage systems are ubiquitous. As classic erasure correction codes fail to scale with the exponential growth of data, regenerating codes were proposed [31].

A regenerating code is described by the parameters \((n, k, d, B, q, \alpha, \beta)\), where \(k \leq d \leq n - 1\) and \(\beta \leq \alpha\). The file \(x \in \mathbb{F}_q^B\) is to be stored on \(n\) storage nodes. The reconstruction degree \(k\) is the number of nodes required to restore \(x\), a process which is called reconstruction, and carried out by a data collector. The repair degree \(d\) is the number of helper nodes which are required to restore a lost node, a process which is called repair, and carried out by a newcomer node (abbrv. newcomer). The parameter \(\alpha\) denotes the number of field elements per storage node, and the parameter \(\beta\) denotes the number of field elements which are to be downloaded from each helper node during repair. Further requirements are the ability to reconstruct from any set of \(k\) nodes, and to repair from any set of \(d\) nodes.

In [31], the parameters of any regenerating code were shown to satisfy the so called cut-set bound

\[
B \leq \sum_{i=0}^{k-1} \min\{\alpha, (d-i)\beta\}, \quad (9.1)
\]

from which a tradeoff between \(\alpha\) and \(\beta\) is apparent. One point of this tradeoff, in which \(\alpha\) is minimized, attains \(\alpha = \frac{B}{k}\), whereas the second point, in which \(\beta\) is minimized, attains \(\alpha = \beta d\). Codes which attain \((9.1)\) with equality and have \(\alpha = \frac{B}{k}\) are called Minimum Storage Regenerating (MSR) codes. Codes which attain \((9.1)\) with equality and have \(\alpha = d\beta\) are called Minimum Bandwidth Regenerating (MBR) codes.

In the first part of this paper, regenerating codes with \(\alpha = d\beta\) are constructed. These codes have \(B\) that asymptotically attains \((9.1)\) with equality as \(k\) increases, and is close to attaining equality even for small values of \(k\). In addition, as long as the file size is large enough, these codes may be realized over any given field, and in particular, the binary field. This restriction on the file size is usually satisfied in typical distributed storage systems.

The second part of the paper contains a construction of regenerating

thesis performed at the Technion, under the supervision of Prof. Tuvi Etzion. e-mail: netanel.raviv@gmail.com.
codes with $d \geq 2k - 2$ that have $B$ which approaches $\alpha k$ as $k$ increases. As in the first part, these codes may also be realized over any given field if the file size is large enough.

The last part presents an alternative approach for the construction in the first part. This approach employs newly studied algebraic objects called subspace codes, which recently gained increasing attention due to their application in error correction for random network coding [13]. This approach does not result in an improvement of the first part of the paper, and yet it presents applications of mathematical objects of independent interest, and a potential for future improvements of the results.

Our techniques are inspired by the code construction in [24], which attains MBR codes for all parameters $n, k$, and $d$, as well as MSR codes for all parameters $n, k$, and $d$ such that $d \geq 2k - 2$, where both constructions have $\beta = 1$. It is noted in [24, Sec. I.C] that only the case $\beta = 1$ is discussed since striping of data is possible, and larger $\beta$ may be obtained by code concatenation. It will be shown in the sequel that allowing a larger $\beta$, not through concatenation, enables a significant reduction in field size with a small and often negligible loss of code rate.

According to [24, Sec. I.B.], regenerating codes which do not attain (9.1) with equality are not MBR codes, even if they satisfy $\alpha = d \beta$ and attain (9.1) asymptotically. Similarly, regenerating codes which attain $B = \alpha k$ asymptotically are not considered MSR codes. To the best of our knowledge, such codes were not previously studied, and hence, we coin the following terms.

**Definition 44** A regenerating code is called a nearly MBR (NMBR) code if it satisfies $\alpha = \beta d$, and $B$ approaches the cut-set bound (9.1) as $k$ increases. Similarly, a regenerating code is called a nearly MSR (NMSR) code if it has $B$ which approaches $\alpha k$ as $k$ increases.

This paper is organized as follows. Previous work is discussed in Section 9.A.2. Mathematical background on several notions from field theory, number theory, and matrix analysis is given in Section 9.A.3. NMBR codes are given in Section 9.A.4 and NMSR codes in Section 9.A.5, each of which contains a subsection with a detailed asymptotic analysis and numerical examples. The aforementioned connection to subspace codes is given in Section 9.A.6 and concluding remarks with future research directions are
given in Section 9.A.7. Finally, omitted proofs and additional properties of
the given codes are given in the Appendices.

9.A.2 Previous Work

MBR codes for all parameters \(n, k,\) and \(d\) were constructed in \[24\], where the
underlying field size \(q\) must be at least \(n\), and \(B = \left(\frac{k+1}{2}\right) + k(d - k)\). MSR
codes for all parameters \(n, k,\) and \(d\) such that \(d \geq 2k - 2\) were also constructed
in \[24\], where the underlying field size must be at least \(n(d - k + 1)\), and
\(B = (d - k + 1)(d - k + 2)\). These codes are given under a powerful framework
called product matrix codes, and are the main objects of comparison in this
paper. Henceforth, these codes are denoted by PM-MBR and PM-MSR,
respectively.

Broadly speaking, the construction of PM-MBR codes associates a dis-
tinct field element \(\gamma\) to each storage node, which stores \((1, \gamma, \gamma^2, \ldots, \gamma^d) \cdot M,\)
where \(M\) is a symmetric matrix that contains the file \(x\). In our NMBR
construction we replace the vector \((1, \gamma, \gamma^2, \ldots, \gamma^d)\) by a properly chosen
matrix. This matrix is associated with an element of an extension field of
the field \(\mathbb{F}_q\), an approach which enables a reduction in field size. Our NMSR
construction uses similar notions, where the proofs are a bit more involved,
and require tools from basic number theory and matrix analysis.

Product matrix codes were recently improved in \[7\]. The improvement is
obtained by operating over a ring \(R_m\) in which addition and multiplication
may be implemented by cyclic shifts and binary additions. The size of \(R_m\)
is \(2^m\), where \(m\) must not be divisible by \(2, \ldots, n - 1\) \[7, Th. 10\]. Using
our techniques, it is possible to employ the binary field itself (or any other
given field), rather than the aforementioned ring \(R_m\), with minor loss of
rate. PM-MSR codes were also recently improved in \[18\], which reduced the
required field size to \(q > n\), whenever \(q\) is a power of two. This result follows
from a special case of our construction (see Remark \[14\]).

A closely related family of MBR codes called repair-by-transfer codes is
discussed in \[14\]. In repair-by-transfer codes a node which participates in a
repair must transfer its data without any additional computations. In \[14\],
the field size required for repair-by-transfer MBR codes is reduced to \(O(n)\)
instead of \(O(n^2)\) in previous constructions \[29\]. A similar result is obtained
by \[17\], which also studied the repair and reconstruction complexities. In
addition, [14] obtain binary repair-by-transfer codes for the special cases

\( k = d = n - 2 \) and \( k + 1 = d = n - 2 \).

Further aspects of regenerating codes where thoroughly studied in recent years [9, 30, 10, 11, 2, 21, 25]. In particular, the problem of constructing high rate MSR codes, i.e., with a constant number of parity nodes, has received a great deal of attention [33, 34, 5, 28, 26]. Implementing our techniques for high rate MSR codes is one of our future research directions.

9.A.3 Preliminaries

This section lists several notions from field theory, linear algebra, number theory and matrix analysis, which are required for the constructions that follow. To this end, the following notations are introduced. For an integer \( m \), the notation \( \mathbb{Z}_m \) stands for the ring of integers modulo \( m \), and \([m] \triangleq \{1, \ldots, m\}\). The ring of univariate polynomials over \( \mathbb{F}_q \) is denoted by \( \mathbb{F}_q[x] \). For integers \( s \) and \( t \), the notations \( \mathbb{F}_{q^{st}} \) stands for the ring of \( s \times t \) matrices over \( \mathbb{F}_q \). For a matrix \( A \), let \( A_{i,j} \) be its \((i,j)\)-th entry, and let \( A_i \) be its \( i \)-th row or column, where ambiguity is resolved if unclear from context. If \( A \) is a matrix in \( \mathbb{F}_{q^{st}}^{m \times m} \) which consists of \( s \cdot t \) blocks of size \( m \times m \) each, we denote its \((i,j)\)-th block by \([A]_{i,j}^{(m)}\), and omit the notation \((m)\) if it is clear from the context. The notations \( I_m \) and \( 0_m \) are used to denote the identity and zero matrix of order \( m \), respectively.

9.A.3.1 Companion matrices and representation of extension fields

**Definition 45** The companion matrix of a monic univariate polynomial

\( P(x) = p_0 + p_1 x + \ldots + p_{e-1} x^{e-1} + x^e \in \mathbb{F}_q[x] \) is the \( e \times e \) matrix

\[
\begin{pmatrix}
0 & 0 & \cdots & -p_0 \\
1 & 0 & \cdots & -p_1 \\
0 & \cdots & \cdots & \ddots \\
0 & \cdots & 1 & -p_{e-1}
\end{pmatrix}.
\]

It is an easy exercise to show that the minimal and characteristic polynomials of a companion matrix are its corresponding polynomial, and the
eigenvalues are the roots of that polynomial (which may reside in an extension field of the field of coefficients).

The following lemma, which is well-known, provides a convenient yet redundant representation of extension fields as matrices over the base field. Unlike other representations, this representation encapsulates both the additive and the multiplicative operations in the extension field, both as the respective operations between matrices.

Lemma 127 [16, Ch. 2, Sec. 5] If \( P \in \mathbb{F}_q[x] \) is monic and irreducible of degree \( m \) with companion matrix \( M_P \), then the linear span over \( \mathbb{F}_q \) of the set \( \{M_P^i\}_{i=0}^{m-1} \) is isomorphic to \( \mathbb{F}_q^m \). If \( P \) is also primitive, then \( \{M_P^i\}_{i=0}^{q^m-2} \cup \{0\} \) is isomorphic to \( \mathbb{F}_q^m \).

Lemma 127 also has an inverse [32]. That is, given the field \( \mathbb{F}_q^m \), it is possible to represent its elements as all powers of the companion matrix \( P \) which corresponds to an irreducible polynomial of degree \( m \) over \( \mathbb{F}_q \). Hence, for any \( m \) and any such matrix \( P \), let \( \theta_P : \mathbb{F}_q^m \to \mathbb{F}_q^{m \times m} \) be the function which maps an element in the extension field \( \mathbb{F}_q^m \) to its matrix representation in \( \mathbb{F}_q^{m \times m} \) as a linear combination of powers of \( P \), and since our results are oblivious to the choice of \( P \), we denote \( \theta_P \) by \( \theta \). Notice that \( \theta \) is a field isomorphism, that is, every \( y_1 \) and \( y_2 \) in \( \mathbb{F}_q^m \) satisfy that \( \theta(y_1 \cdot y_2) = \theta(y_1) \cdot \theta(y_2) \) and \( \theta(y_1 + y_2) = \theta(y_1) + \theta(y_2) \). The function \( \theta \) can be naturally extended to matrices, where \( A \in \mathbb{F}_q^{s \times t} \) is mapped to

\[
\Theta(A) \triangleq \begin{pmatrix}
\theta(A_{1,1}) & \theta(A_{1,2}) & \cdots & \theta(A_{1,t}) \\
\theta(A_{2,1}) & \theta(A_{2,2}) & \cdots & \theta(A_{2,t}) \\
\vdots & \vdots & \ddots & \vdots \\
\theta(A_{s,1}) & \theta(A_{s,2}) & \cdots & \theta(A_{s,t})
\end{pmatrix} \in \mathbb{F}_q^{ms \times mt}. \tag{9.2}
\]

Lemma 128 For any integers \( m, s, t \) and \( \ell \), if \( A \in \mathbb{F}_q^{s \times t} \) and \( B \in \mathbb{F}_q^{t \times \ell} \) then \( \Theta(AB) = \Theta(A) \cdot \Theta(B) \).
**Proof.** By the definition of $\Theta$, and by using the fact that $\theta$ is a field isomorphism, for all $i \in [s]$ and $j \in [\ell]$ we have that

$$
[\Theta(AB)]_{i,j}^{(m)} = \theta \left( \sum_{k=1}^{t} A_{i,k} B_{k,j} \right) = \sum_{k=1}^{t} \theta(A_{i,k}) \theta(B_{k,j})
$$

$$
= \sum_{k=1}^{t} [\Theta(A)]_{i,k}^{(m)} [\Theta(B)]_{k,j}^{(m)} = [\Theta(A) \cdot \Theta(B)]_{i,j}^{(m)}.
$$

\[\square\]

**Lemma 129** For any integers $m$ and $t$, if $A \in \mathbb{F}_{q^m}^{t \times t}$ is invertible then $\Theta(A) \in \mathbb{F}_{q^m}^{mt \times mt}$ is invertible.

**Proof.** According to Lemma 128 since $A^{-1}$ exists it follows that

$$I_{mt} = \Theta(I_t) = \Theta(A \cdot A^{-1}) = \Theta(A) \cdot \Theta(A^{-1}),$$

and hence $\Theta(A^{-1})$ is the inverse of $\Theta(A)$.

\[\square\]

**9.A.3.2 Kronecker products and cyclotomic cosets**

The proofs of the construction of NMSR codes in Subsection 9.A.5.1 are slightly more involved than those given in other sections. The main tools in those proofs are cyclotomic cosets and Kronecker products, which are discussed in this subsection.

**Definition 46** [8 Sec. 3.7], [27 Sec. 7.5] For an integer $m$, a prime power $q$ such that $\gcd(q,m) = 1$, and $s \in \mathbb{Z}_m$, a subset of $\mathbb{Z}_m$ of the form \{s, sq, sq^2, sq^3, \ldots\} is called a $q$-cyclotomic coset modulo $m$.

It is well known (e.g., [27, 8]) that for any $m$ such that $\gcd(q,m) = 1$, the size of any $q$-cyclotomic coset modulo $m$ divides the order of $q$ in $\mathbb{Z}_m$ (that is, the smallest integer $t$ such that $q^t \equiv 1 \pmod{m}$).
Definition 47 For a matrix $A \in \mathbb{F}_q^{s \times t}$ and a matrix $B \in \mathbb{F}_q^{n \times m}$, the Kronecker product $A \otimes B$ is the matrix

$$
\begin{pmatrix}
A_{1,1}B & A_{1,2}B & \cdots & A_{1,t}B \\
A_{2,1}B & A_{2,2}B & \cdots & A_{2,t}B \\
\vdots & \vdots & \ddots & \vdots \\
A_{s,1}B & A_{s,2}B & \cdots & A_{s,t}B
\end{pmatrix} \in \mathbb{F}_q^{sn \times tm}.
$$

The Kronecker product is useful when solving equations in which the unknown variable is a matrix. This application is enabled through an operator called vec, defined as follows.

Definition 48 For a matrix $A \in \mathbb{F}_q^{s \times t}$ with columns $A_1, \ldots, A_t$, let

$$
\text{vec}(A) \triangleq \begin{pmatrix}
A_1 \\
A_2 \\
\vdots \\
A_t
\end{pmatrix} \in \mathbb{F}_q^{st}.
$$

The following two lemmas present several properties of the Kronecker product and the vec operator. These lemmas are well known, and their respective proofs may be found, e.g., in [15, 1, 20]. In particular, Lemma 130 which follows discusses a close variant of the so called Sylvester equation $AX + XB = C$, where $A, B,$ and $C$ are known matrices, and $X$ is an unknown matrix. For completeness, full proofs are detailed below.

Lemma 130 For an integer $m$, if $A, X,$ and $B$ are $m \times m$ matrices over $\mathbb{F}_q$, then $\text{vec}(AXB - X) = (B^\top \otimes A - I_m) \cdot \text{vec}(X)$. 

295
Proof. Clearly, if $X_1, \ldots, X_m$ are the columns of $X$ and $B_1, \ldots, B_m$ are the columns of $B$, then the $i$-th column of $(AXB - X)$ is

$$AXB_i - X_i = \sum_{j=1}^{m} B_{j,i}(AX)_j - X_i$$
$$= \sum_{j=1}^{m} B_{j,i}AX_j - X_i$$

$$= \left( B_{1,i}A \ B_{2,i}A \ \ldots \ B_{i-1,i}A \ B_{i,i}A - I_m \ B_{i+1,i}A \ \ldots \ B_{m,i}A \right) \cdot \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix},$$

and hence, according to Definition 47, it follows that

$$\text{vec}(AXB - X) = \left( B_{1,1}A - I_m \ B_{2,1}A \ \ldots \ B_{m,1}A \ B_{1,2}A \ B_{2,2}A - I_m \ \ldots \ B_{m,2}A \ \vdots \ \vdots \ \vdots \ B_{1,m}A \ B_{2,m}A \ \ldots \ B_{m,m}A - I_m \right) \cdot \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_m \end{pmatrix}$$

$$= \left( B^\top \otimes A - I_{m^2} \right) \cdot \text{vec}(X).$$

Lemma 131 If $A$ and $B$ are two $m \times m$ matrices over $\mathbb{F}_{q'}$ and $\mathbb{F}_{q'}$ is a field which contains all eigenvalues $\lambda_1, \ldots, \lambda_m$ of $A$ and $\mu_1, \ldots, \mu_m$ of $B$, then the eigenvalues of $A \otimes B$ are $\{\lambda_i \mu_j | i, j \in [m] \}$.

Proof. For any $i$ and $j$ in $[m]$, let $v_i$ and $u_j$ be (column) eigenvectors in $\mathbb{F}_{q'}^m$ such that $Av_i = \lambda_i v_i$ and $Bu_j = \mu_j u_j$. By Definition 47, it follows
that

\[(A \otimes B) \cdot (u_j \otimes v_i) = \begin{pmatrix}
B_{1,1}A & B_{1,2}A & \cdots & B_{1,m}A \\
B_{2,1}A & B_{2,2}A & \cdots & B_{2,m}A \\
\vdots & \vdots & \ddots & \vdots \\
B_{m,1}A & B_{m,2}A & \cdots & B_{m,m}A \\
\end{pmatrix} \cdot \begin{pmatrix}
u_{j,1}v_i \\
u_{j,2}v_i \\
\vdots \\
u_{j,n}v_i \\
\end{pmatrix} = \begin{pmatrix}
u_{j,1}B_{1,1}Av_i + \nu_{j,2}B_{1,2}Av_i + \cdots + \nu_{j,n}B_{1,m}Av_i \\
u_{j,1}B_{2,1}Av_i + \nu_{j,2}B_{2,2}Av_i + \cdots + \nu_{j,n}B_{2,m}Av_i \\
\vdots \\
u_{j,1}B_{m,1}Av_i + \nu_{j,2}B_{m,2}Av_i + \cdots + \nu_{j,n}B_{m,m}Av_i \\
\end{pmatrix} = \lambda_i \cdot \begin{pmatrix}
u_{j,1}B_{1,1}u_jv_i + \nu_{j,2}B_{1,2}u_jv_i + \cdots + \nu_{j,n}B_{1,m}u_jv_i \\
u_{j,1}B_{2,1}u_jv_i + \nu_{j,2}B_{2,2}u_jv_i + \cdots + \nu_{j,n}B_{2,m}u_jv_i \\
\vdots \\
u_{j,1}B_{m,1}u_jv_i + \nu_{j,2}B_{m,2}u_jv_i + \cdots + \nu_{j,n}B_{m,m}u_jv_i \\
\end{pmatrix} = \lambda_i (Bu_j) \otimes (v_i) = \lambda_i \mu_j (u_j \otimes v_i).\]
9.A.4 Nearly MBR codes

For any given $n, k, d, q$, and a sufficiently large file size $B$, this section presents regenerating codes with $\alpha = d\beta$, and $B$ which approaches the cut-set bound as $k$ increases. For any such $n, k, d$ and $q$ let $b$ be an integer such that

A1. $b \geq k \log_q n,$

A2. $k \mid b,$

and let $B \triangleq \frac{b(b+1)}{2} + b^2 \left(\frac{d}{k} - 1\right).$ Notice that Condition A1 implies that $B = \Omega \left( kd \log_q (n)^2 \right)$. Since usually, the file size $B$ is in the order of magnitude of billions, and the number of nodes is in the order of magnitude of dozens, Condition A1 is trivially satisfied in many distributed storage systems (see Section 9.A.4.2 for explicit examples).

9.A.4.1 Construction

Given a file $x \in \mathbb{F}_q^B$, define the following data matrix, which resembles the one in [24]:

$$
X = \begin{pmatrix}
S & T \\
T^\top & 0
\end{pmatrix} \in \mathbb{F}_q^{\frac{db}{k} \times \frac{db}{k}},
$$

where

$$
S \triangleq \begin{pmatrix}
x_1 & x_2 & x_3 & \ldots & x_b \\
x_2 & x_{b+1} & x_{b+2} & \ldots & x_{2b-1} & \vdots & \vdots & \vdots & \vdots \\
x_b & \cdots & \cdots & \cdots & \cdots
\end{pmatrix} \in \mathbb{F}_q^{b \times b}, \quad (9.3)
$$

and $T \in \mathbb{F}_q^{b \times \left(\frac{d}{k} - 1\right)}$ contains the remaining $b^2 \left(\frac{d}{k} - 1\right)$ elements of $x$ in some arbitrary order.

Let $P$ be a companion matrix of any primitive polynomial of degree $\frac{b}{k}$ over $\mathbb{F}_q$, and let $i_1, \ldots, i_n$ be distinct integers in the range $\{0, \ldots, q^{b/k} - 1\}$, which exist by A1. Using $P$ and $i_1, \ldots, i_n$, define the following encoding matrix,
\[
M = \begin{pmatrix}
  M_1 \\
  M_2 \\
  \vdots \\
  M_n
\end{pmatrix} \in \mathbb{F}_q^{\frac{n b}{k} \times \frac{d b}{k}}, \text{ where}
\]

\[
M_j \triangleq \begin{pmatrix} I & P^{i_j} & P^{2i_j} & \ldots & P^{(d-1)i_j} \end{pmatrix} \in \mathbb{F}_q^{\frac{b}{k} \times \frac{d b}{k}}, \quad (9.4)
\]

and store \( M_j \cdot X \) in storage node \( j \). Notice that by the definition of the matrix \( P \), we have that \( \alpha = \frac{b^2}{k^2} \cdot d \).

**Remark 12** It is possible to replace \( P \) by the companion matrix of an irreducible polynomial which is not necessarily primitive, in which case, let \( \{A_j(P)\}^{n}_{j=1} \) be distinct nonzero linear combinations of \( \{P^i\}^{b/k-1}_{i=0} \), and define \( M_j \triangleq \begin{pmatrix} I & A_j(P) & A_j(P)^2 & \ldots & A_j(P)^{d-1} \end{pmatrix} \). However, we choose a primitive polynomial for convenience.

**Remark 13** For \( k = b \) this code is a PM-MBR code [24, Sec. IV], and in which case Condition A1 implies that \( q \geq n \). Therefore, the advantage of our techniques exists only for \( b > k \).

**Theorem 51** In the above code, exact repair of any failed node may be achieved by downloading \( \beta \triangleq \frac{b^2}{k^2} \) field elements from any \( d \) of the remaining nodes.

**Proof.** Assume that node \( i \) failed, and \( D = \{j_1, \ldots, j_d\} \) is a subset of \( [n] \) of size \( d \) such that \( i \notin D \). To repair node \( i \), every node \( j_i \in D \) computes \( M_{j_i}XM_i^\top \), which is a \( \frac{b}{k} \times \frac{b}{k} \) matrix over \( \mathbb{F}_q \), and sends it to the newcomer. The newcomer obtains

\[
\begin{pmatrix}
  M_{j_1}XM_i^\top \\
  M_{j_2}XM_i^\top \\
  \vdots \\
  M_{j_d}XM_i^\top
\end{pmatrix} = \begin{pmatrix}
  M_{j_1} \\
  M_{j_2} \\
  \vdots \\
  M_{j_d}
\end{pmatrix} \cdot X \cdot M_i^\top \triangleq M_DXM_i^\top.
\]
According to [9.4], the matrix $M_D$ is of the form

$$M_D = \begin{pmatrix}
    I & P^{i_{j_1}} & P^{2i_{j_1}} & \ldots & P^{(d-1)i_{j_1}} \\
    I & P^{i_{j_2}} & P^{2i_{j_2}} & \ldots & P^{(d-1)i_{j_2}} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    I & P^{i_{j_d}} & P^{2i_{j_d}} & \ldots & P^{(d-1)i_{j_d}}
\end{pmatrix}.$$

Since $M_D$ can be written as $\Theta(M'_D)$ for some invertible Vandermonde matrix $M'_D$ in $\mathbb{F}_q^{d\times d}$, it follows by Lemma 129 that $M_D$ is invertible. Thus, the newcomer may multiply from the left by $M^{-1}_D$ and obtain $XM_i^T$. Since $X$ is a symmetric matrix, exact repair is obtained by transposing. ■

**Theorem 52** In the above code, reconstruction may be achieved by downloading $\alpha = \frac{b^2}{k} \cdot d$ field elements per node from any $k$ nodes.

**Proof.** Let $K = \{j_1, \ldots, j_k\}$ be a subset of $[n]$ of size $k$, and download $M_{j_i}X$ from node $j_i$ for each $j_i \in K$. The data collector thus obtains

$$M_K \cdot X \triangleq \begin{pmatrix}
    I & P^{i_{j_1}} & P^{2i_{j_1}} & \ldots & P^{(d-1)i_{j_1}} \\
    I & P^{i_{j_2}} & P^{2i_{j_2}} & \ldots & P^{(d-1)i_{j_2}} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    I & P^{i_{j_k}} & P^{2i_{j_k}} & \ldots & P^{(d-1)i_{j_k}}
\end{pmatrix} \begin{pmatrix}
    S \\
    T^T \\
    0
\end{pmatrix} = (M'_K S + M''_K T^T, M'_K T),$$

where

$$M'_K \triangleq \begin{pmatrix}
    I & P^{i_{j_1}} & P^{2i_{j_1}} & \ldots & P^{(k-1)i_{j_1}} \\
    I & P^{i_{j_2}} & P^{2i_{j_2}} & \ldots & P^{(k-1)i_{j_2}} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    I & P^{i_{j_k}} & P^{2i_{j_k}} & \ldots & P^{(k-1)i_{j_k}}
\end{pmatrix},$$

and

$$M''_K \triangleq \begin{pmatrix}
    P^{ki_{j_1}} & P^{2i_{j_1}} & \ldots & P^{(d-1)i_{j_1}} \\
    P^{ki_{j_2}} & P^{2i_{j_2}} & \ldots & P^{(d-1)i_{j_2}} \\
    \vdots & \vdots & \ddots & \vdots \\
    P^{ki_{j_k}} & P^{2i_{j_k}} & \ldots & P^{(d-1)i_{j_k}}
\end{pmatrix}. $$
As in the proof of Theorem 51, we have that $M'_{K}$ is invertible. Hence, if $d > k$, the matrix $T$ may be restored by extracting the rightmost $b^2 \left(\frac{d}{k} - 1\right)$ columns of $M_K X$ and multiplying by $(M'_{K})^{-1}$. Having $T$, it can be used to reduce $M''_K T^\top$ from the remaining columns of $M_K X$, and then extracting $S$ is similar. If $d = k$, then $X = S$, and multiplication by $M_K = M'_{K}$ suffices for reconstruction. ■

By Theorem 51 and Theorem 52 it is evident that $\alpha = d\beta$, and hence this construction attains minimum bandwidth repair. In Subsection 9.A.4.2 it will be shown that although the cut-set bound is not attained with equality, $B$ approaches the cut-set bound (9.1) as $k$ increases. Moreover, it will be evident that a small and often negligible loss of rate is obtained already for small values of $k$.

### 9.A.4.2 The proximity of NMBR codes to MBR codes

In this section it is shown that the codes constructed in Section 9.A.4.1 do not attain the cut-set bound (9.1), and hence cannot be considered MBR codes even though they attain $\alpha = d\beta$ (see Definition 44 and its preceding discussion). However, it is also shown that the cut-set bound is nearly achieved for large enough $k$, together with few specific examples which demonstrate a small loss of rate.

Let $C \triangleq \sum_{i=0}^{k-1} \min\{\alpha, (d - i)\beta\}$, and recall that by (9.1) we have that $B \leq C$ for all regenerating codes. Clearly, for codes which attain $\alpha = d\beta$ we have that

$$C = \sum_{i=0}^{k-1} \min\{d\beta, (d - i)\beta\} = \beta \sum_{i=0}^{k-1} (d - i) = \beta \left( dk - \frac{k(k - 1)}{2} \right). \quad (9.5)$$

Hence, for the codes which are presented in Section 9.A.4.1 we have that $C = \frac{b^2}{k} \left( d - \frac{k-1}{2} \right)$. It is readily verified that indeed, $C > B$, thus (9.1) is not attained, and hence these are not MBR codes. However, we have that

$$\frac{B}{C} = \frac{2d - k \left( 2 - \frac{b+1}{b} \right)}{2d - k + 1} = \frac{2 - \frac{k}{a} \cdot \frac{b-1}{b}}{2 - \frac{k}{a} + \frac{1}{a}}$$

301
and hence the cut-set bound is achieved in the asymptotic regime. That is, since a large $k$ implies a large $b$ (since $k|b$ in Condition A2) and a large $d$ (since $d \geq k$), by following the outline of Section 9.A.4.1 and choosing a large enough $k$, one may obtain a code in which $B$ is arbitrarily close to $C$, regardless of the relation between $k$ and $d$.

In the remainder of this section, a detailed comparison of parameters between the PM-MBR codes and our NMBR codes is given. From these examples it will be evident that the decrease in file size (in comparison with the cut-set bound), and hence the decrease in the code rate, is a small price to pay for a considerable reduction in field size.

The curious reader might suggest that the extension field representation which is given in Lemma 127, can be applied directly to PM-MBR codes over an extension field, obtaining regenerating codes over the respective base field. This intuition is formalized in the following definition. For this definition, recall that PM-MBR codes may be obtained by choosing $k = b$ in the construction in Subsection 9.A.4.1.

**Definition 49** Given a PM-MBR code over an extension field $\mathbb{F}_{q^m}$ with an encoding matrix $M$ and data matrix $X$, let EPM-MBR be the code over $\mathbb{F}_q$ which results from applying the function $\Theta$ from (9.2) on the encoding matrix $M$ and multiplying it by a data matrix $X'$. The data matrix $X'$ is given by applying $\theta$ on the upper triangular part of the data matrix $X$, and completing the lower triangular part to obtain symmetry.

In order to apply the repair and reconstruction algorithms from Theorem 51 and Theorem 52 to EPM-MBR codes, the data matrix $X'$ must be symmetric. Hence, it follows that the only data matrices $X \in \mathbb{F}_{q^m}^{d \times d}$ on which EPM-MBR codes maintain their repair and reconstruction capabilities are those in which all diagonal submatrices $\theta(X_{i,i})$ of $X'$ are symmetric. Since companion matrices are in general not symmetric, this usually induces a further loss of rate. For simplicity, we shall ignore this detail in the comparison which follows, since NMBR codes will be shown to supersede EPM-MBR codes even without this additional rate loss. In the remainder of this section we compare between EPM-MBR codes, NMBR codes, and PM-MBR codes with the concise vector space representation of extension field elements.\footnote{I.e., each extension field element is represented by a vector over the base field.}
In PM-MBR codes the file size $B$ is a function of $k$ and $d$, and in addition, $\beta = 1$. Further, all parameters are measured in field elements rather than in bits. Therefore, to achieve a fair comparison, one must concatenate a PM-MBR code to itself in order to obtain the same parameters $n, k, d, \alpha$, and $\beta$ when measured in bits, and only then compare the resulting $B, q$, and the rate $\frac{B}{n \alpha}$. In addition, since fields of even characteristic are essential for hardware implementation, we restrict our attention to $q = 2$ in our codes, and to $q$ which is an integer power of 2 for PM-MBR codes. Hence, the PM-MBR code is concatenated with itself $b \cdot 2^{\lceil \log n \rceil} k^2$ times, and considered with the same $q = 2^{\lceil \log n \rceil}$, where each element in this field is represented by a vector in $\mathbb{F}_2^{\lceil \log n \rceil \cdot k^2}$. Similarly, the EPM-MBR code is concatenated with itself $b \cdot 2^{\lceil \log n \rceil} k^2$ times, and considered with the same $q = 2^{\lceil \log n \rceil}$, where each element in this field is represented by a square matrix in $\mathbb{F}_2^{\lceil \log n \rceil \times \lceil \log n \rceil}$.

Notice that MBR codes have $B = \beta (dk - k(k-1)/2)$ (see (9.3)), where $B$ is measured in elements over $\mathbb{F}_q$. Therefore, by setting $\beta = 1$, $q = 2^{\lceil \log n \rceil}$, and concatenating a PM-MBR code $\frac{b^2}{\lceil \log n \rceil k^2}$ times with itself, we have that the number of information bits in the file is $C = \frac{b^2}{k} (d - \frac{k-1}{2})$. Similarly, by concatenating an EPM-MBR code $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ times with itself, since each field element is represented by a $\lceil \log n \rceil \times \lceil \log n \rceil$ binary matrix that contains $\lceil \log n \rceil$ information bits, it follows that the number of information bits in the file is $\beta (dk - k(k-1)/2) \cdot \frac{b^2}{\lceil \log n \rceil^2 k^2} \cdot \lceil \log n \rceil = \frac{C}{\lceil \log n \rceil}$. As a result, by fixing any $n, k$, and $d$ such that $k \leq d \leq n - 1$, we have Table 9.1 in which the values of $\beta, \alpha$, and $B$ are given in bits.

Table 9.2 contains specific examples of the comparison given in Table 9.1. The parameter $b$ is chosen such that $\frac{b^2}{\lceil \log n \rceil k^2}$ and $\frac{b^2}{\lceil \log n \rceil^2 k^2}$ are integers, and such that the resulting file size is within one of several common use cases. Notice that much smaller values of $b$ may be chosen, for example, if one wishes to increase concurrency by code concatenation. For convenience, some values are given in either MegaBytes (MB), GigaBytes (GB), or Terabytes (TB) rather than in bits.

From Table 9.2 it is evident that in comparison with PM-MBR codes, a considerable reduction in field size is obtained by our codes, even for rather small values of $k$. Furthermore, our techniques obtain a larger rate in comparison with EPM-MBR codes, which are implemented over the binary
In many practical applications [23, Slide 38], multiplication in a finite field $F_{2^w}$ is implemented by table look-ups for $w \leq 8$, and sometimes considered infeasible in large systems with $w > 8$, since it requires numerous table look-ups and expensive arithmetic. Hence, for $n > 2^8 = 256$, our techniques improve the feasibility of storage codes without compromising the code rate significantly.

### 9.A.5 Nearly MSR codes

In this section, for any given $n, k, d, q$ such that $d \leq n - 1$ and $d = 2k - 2$, and for a sufficiently large file size $B$, regenerating codes in which $B$ approaches $\alpha k$ as $k$ increases are provided. Codes for $d > 2k - 2$ with similar properties are obtained in the sequel from this construction. For any such $n, k, d$ and $q$, let $b$ be an integer such that

1. $n \leq \frac{q^{b/k}-1}{g^b}$, where $g \triangleq \gcd(k - 1, q^{b/k} - 1)$,
2. $k \mid b$.
Table 9.2: A comparison of parameters between our NMBR codes (Subsection 9.A.4.1) and the PM-MBR codes [24, Sec. IV] for several common parameters $n, k, d$.

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>$k$</th>
<th>$d$</th>
<th>$\alpha$</th>
<th>$\beta$</th>
<th>$b$</th>
<th>$q$</th>
<th>$B$</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMBR</td>
<td>30</td>
<td>20</td>
<td>20</td>
<td>250MB</td>
<td>12.5MB</td>
<td>$10000 \cdot k$</td>
<td>2</td>
<td>$\approx 2.5$GB</td>
<td>$\approx 0.33$</td>
</tr>
<tr>
<td>PM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td>$2.625$GB</td>
<td>0.35</td>
</tr>
<tr>
<td>EPM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0.525GB</td>
<td>0.07</td>
</tr>
<tr>
<td>NMBR</td>
<td>26</td>
<td>22</td>
<td>24</td>
<td>$\approx 841.7$MB</td>
<td>$\approx 35.07$MB</td>
<td>$16750 \cdot k$</td>
<td>2</td>
<td>$\approx 10.03$GB</td>
<td>$\approx 0.4583$</td>
</tr>
<tr>
<td>PM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>32</td>
<td>$\approx 10.41$GB</td>
<td>$\approx 0.4759$</td>
</tr>
<tr>
<td>EPM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2.8GB</td>
<td>0.095</td>
</tr>
<tr>
<td>NMBR</td>
<td>260</td>
<td>220</td>
<td>240</td>
<td>$\approx 8.416$GB</td>
<td>$\approx 35.06$MB</td>
<td>$16749 \cdot k$</td>
<td>512</td>
<td>$\approx 1.006$TB</td>
<td>$\approx 0.4601$</td>
</tr>
<tr>
<td>PM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>$\approx 0.11$TB</td>
<td>$\approx 0.05$</td>
</tr>
<tr>
<td>EPM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>0.11TB</td>
<td>0.05</td>
</tr>
<tr>
<td>NMBR</td>
<td>2600</td>
<td>2200</td>
<td>2400</td>
<td>$\approx 84.18$GB</td>
<td>$\approx 35.07$MB</td>
<td>$16752 \cdot k$</td>
<td>4096</td>
<td>$\approx 100.36$TB</td>
<td>$\approx 0.4585$</td>
</tr>
<tr>
<td>PM-MBR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>$\approx 8.36$TB</td>
<td>$\approx 0.038$</td>
</tr>
</tbody>
</table>

and let $B \triangleq \frac{b(k-1)}{k} \cdot \left( \frac{b(k-1)}{k} + 1 \right) = \frac{b^2(k-1)}{k} \left( 1 - \frac{1}{k} + \frac{1}{b} \right)$. Condition [B1] implies that $\frac{nq}{k} \leq \frac{q^{b/k-1}}{b}$, and thus, since $g \leq k - 1$, it follows that any integer $b$ such that $b \geq k(\log_q n + \log_q b)$ suffices. Further, Condition [B1] implies that $B = \Omega \left( k^2(\log_q(n) + \log_q(b))^2 \right)$, and hence it is trivially satisfied in many distributed storage systems.

9.A.5.1 Construction

Similar to [24], given a file $x \in \mathbb{F}_q^B$, arrange its symbols in the upper triangle of two square matrices $S_1, S_2$ of dimensions $\frac{b(k-1)}{k} \times \frac{b(k-1)}{k}$ over $\mathbb{F}_q$, complete
the lower triangle of $S_1, S_2$ to obtain symmetry, and define

$$X \triangleq \begin{pmatrix} S_1 \\ S_2 \end{pmatrix}.$$  

Next, a set of integers $i_1, \ldots, i_n$ in the range $\{0, \ldots, q^{b/k-1} - 1\}$ is chosen such that no two reside in the same $q$-cyclotomic coset modulo $q^{b/k-1}/g$. This choice is enabled by the following lemma.

**Lemma 133** The size of $q$-cyclotomic cosets modulo $q^{b/k-1}/g$ is at most $b/k$.

**Proof.** According to [8, Th. 4.1.4, p. 123], for any $m$, the size of any $q$-cyclotomic coset modulo $m$ is a divisor of $\text{ord}_m(q)$, where $\text{ord}_m(q)$ is the smallest integer $t$ such that $q^t = 1 \pmod{m}$. Since clearly, $q^{b/k-1}/g \mid q^{b/k} - 1$, it follows that $q^{b/k} = 1 \pmod{q^{b/k-1}/g}$, which implies that $\text{ord}_{(q^{b/k-1}/g)}(q)$ is at most $b/k$, and the claim follows. ■

Lemma 133 implies that there are at least $q^{b/k-1} \cdot b/k$ different $q$-cyclotomic cosets modulo $q^{b/k-1}/g$, which enables the choice of $i_1, \ldots, i_n$ by Condition B1. Notice that the choice of $i_1, \ldots, i_n$ is possible using a simple algorithm, which maintains a list of feasible elements, iteratively picks an arbitrary element as the next $i_j$, and removes its coset from the list.

Let $P$ be a companion matrix of any primitive polynomial of degree $b/k$ over $\mathbb{F}_q$, and let

$$\Phi \triangleq \begin{pmatrix} I & P^{i_1} & \ldots & P^{i_1(k-2)} \\ I & P^{i_2} & \ldots & P^{i_2(k-2)} \\ \vdots & \vdots & \ddots & \vdots \\ I & P^{i_n} & \ldots & P^{i_n(k-2)} \end{pmatrix} \in \mathbb{F}_q^{bn \times b(k-1)}$$

and

$$\Lambda \triangleq \begin{pmatrix} P^{i_1(k-1)} \\ P^{i_2(k-1)} \\ \vdots \\ P^{i_n(k-1)} \end{pmatrix} \in \mathbb{F}_q^{bn \times 1}.$$  

Define the $\frac{bn}{k} \times \frac{bd}{k}$ encoding matrix over $\mathbb{F}_q$ as $M \triangleq \begin{pmatrix} \Phi & \Lambda \Phi \end{pmatrix}$ and notice that $M$ is a block-Vandermonde matrix. Moreover, according to Lemma 127.
Lemma [129] and the choice of $i_1, \ldots, i_n$, it follows from the properties of Vandermonde matrices in $\mathbb{F}_q^{n \times d}$ that any $\frac{bd}{k} \times \frac{bd}{k}$ block submatrix of $M$ is invertible. Similarly, every $\frac{b(k-1)}{k} \times \frac{b(k-1)}{k}$ block submatrix of $\Phi$ is invertible. Let $M_i$ be the $i$-th block-row of $M$, and store $M_i \cdot X$ in node $i$. By the definition of the corresponding matrices, we have that $\alpha = b^2 \frac{(k-1)^2}{k^2}$.

Remark 14 For $k = b$ this code is a special case of an PM-MSR code [24, Sec. V], and in which case condition B1 implies that $q \geq n \cdot \gcd(k-1, q-1) + 1$. Hence, the advantage of our techniques exists not only for $b > k$, unlike Remark 13. This improvement also follows from [18, Eq. (37)].

Theorem 53 In the above code, exact repair of any failed node may be achieved by downloading $\beta \triangleq b^2 \frac{(k-1)^2}{k^2}$ field elements from any $d$ of the remaining nodes.

Proof. Assume that node $\ell$ failed and $D = \{j_1, \ldots, j_d\}$ is a subset of $[n]$ of size $d$ such that $\ell \notin D$. Let $\Phi_\ell$ be the $\ell$-th block-row of $\Phi$, and notice that node $\ell$ stored

$$M_\ell X = \left(\Phi_\ell \quad P^{i_\ell (k-1)} \Phi_\ell\right) \cdot X = \Phi_\ell S_1 + P^{i_\ell (k-1)} \Phi_\ell S_2. \quad (9.6)$$

To repair node $\ell$, every node $j_t \in D$ computes $M_{j_t} X \Phi_\ell^\top$, which is a $\frac{k}{k} \times \frac{b}{k}$ matrix over $\mathbb{F}_q$, and sends it to the newcomer. The newcomer obtains

$$\begin{pmatrix}
M_{j_1} X \Phi_\ell^\top \\
M_{j_2} X \Phi_\ell^\top \\
\vdots \\
M_{j_d} X \Phi_\ell^\top
\end{pmatrix} = \begin{pmatrix}
M_{j_1} \\
M_{j_2} \\
\vdots \\
M_{j_d}
\end{pmatrix} \cdot X \cdot \Phi_\ell^\top \triangleq M_D X \Phi_\ell^\top.$$

Since $M_D$ can be seen as $\Theta(M'_D)$ for some full rank Vandermonde matrix $M'_D \in \mathbb{F}_q^d \times d$, it follows from Lemma [129] that $M_D$ is invertible, and hence the newcomer may obtain

$$\left( X \Phi_\ell \right)^\top = \Phi_\ell \cdot \begin{pmatrix} S_1 & S_2 \end{pmatrix},$$

and restore $M_\ell X$ by (9.6). $\blacksquare$

---

3That is, a submatrix which consists of complete blocks.
Theorem 54 In the above code, reconstruction may be achieved by downloading \( \alpha = \frac{b^2(k-1)}{k^2} \) field elements per node from any \( k \) nodes.

Proof. Let \( K = \{j_1, \ldots, j_k\} \) be a subset of \( [n] \) of size \( k \), and download \( M_{j_i}X \) from node \( j_i \) for each \( j_i \in K \). The data collector obtains

\[
M_K X = \Phi_K S_1 + \Lambda_K \Phi_K S_2,
\]

where \( \Lambda_K \) and \( \Phi_K \) are the row-submatrices of \( \Lambda \) and \( \Phi \) which consist of the block-rows which are indexed by \( K \). By multiplying from the right by \( \Phi_K^\top \), the data collector obtains

\[
\Gamma \triangleq \Phi_K S_1 \Phi_K^\top + \Lambda_K \Phi_K S_2 \Phi_K^\top \triangleq W + \Lambda_K Q,
\]

where \( W \) and \( Q \) are symmetric matrices. For \( s \in [k] \) denote \( i_{j_s} \) by \( \ell_s \), and notice that for distinct \( s \) and \( t \) in \( [k] \),

\[
\begin{align*}
[\Gamma]_{s,t} &= [W]_{s,t} + P^{\ell_s(k-1)}[Q]_{s,t} \\
[\Gamma]_{t,s} &= [W]_{t,s} + P^{\ell_t(k-1)}[Q]_{t,s} \\
&= [W]_{s,t}^\top + P^{\ell_t(k-1)}[Q]_{s,t}^\top \\
[\Gamma]_{t,s}^\top &= [W]_{s,t}^\top + [Q]_{s,t} \cdot (P^{\ell_t(k-1)})^\top.
\end{align*}
\] (9.7)

Thus, by subtracting (9.8) from (9.7) we have that

\[
\begin{align*}
[\Gamma]_{s,t} - [\Gamma]_{t,s}^\top &= P^{\ell_s(k-1)}[Q]_{s,t} - [Q]_{s,t} (P^{\ell_t(k-1)})^\top \\
\left([\Gamma]_{s,t} - [\Gamma]_{t,s}^\top\right) \cdot \left(P^{\ell_t(k-1)}\right)^\top &= P^{\ell_s(k-1)}[Q]_{s,t} \left(P^{\ell_t(k-1)}\right)^\top - [Q]_{s,t}.
\end{align*}
\]

Now, it follows from Lemma 130 that vectorizing both sides of this equation results in

\[
\left(P^{\ell_t(k-1)} \otimes P^{\ell_s(k-1)} - I_{b^2/k^2}\right) \cdot \text{vec}([Q]_{s,t}) = \text{vec}\left(\left([\Gamma]_{s,t} - [\Gamma]_{t,s}^\top\right) \cdot \left(P^{\ell_t(k-1)}\right)^\top\right),
\]

which may be seen as a linear system of equations whose variables are the unknown entries of \( [Q]_{s,t} \). According to Lemma 132 this equation has a unique solution if and only if 1 is not an eigenvalue of \( P^{\ell_t(k-1)} \otimes P^{\ell_s(k-1)} \).
Since the characteristic polynomial of any companion matrix is its corresponding polynomial, and since for $P$ this polynomial is primitive, the eigenvalues of $P$ are $\gamma, \gamma^q, \ldots, \gamma^{q^{b/k} - 1}$, where $\gamma$ is some primitive element in $\mathbb{F}_{q^{b/k}}$ [3, Th. 4.1.1, p. 123]. Therefore, the eigenvalues of $P^{\ell_s(k-1)}$ are $\gamma^{\ell_s(k-1)}, \gamma^{\ell_s(k-1)q}, \ldots, \gamma^{\ell_s(k-1)q^{b/k} - 1}$, the eigenvalues of $P^{-\ell_t(k-1)}$ are $\gamma^{-\ell_t(k-1)}, \gamma^{-\ell_t(k-1)q}, \ldots, \gamma^{-\ell_t(k-1)q^{b/k} - 1}$, and by Lemma 3.13 the eigenvalues of $P^{-\ell_t(k-1)} \otimes P^{\ell_s(k-1)}$ are

$$\Delta \triangleq \left\{ \gamma^{\ell_s(k-1)q^e - \ell_t(k-1)q^h} \mid e, h \in \{0, 1, \ldots, b/k - 1\} \right\}.$$ 

If $1 \in \Delta$, it follows that there exist $e$ and $h$ in $\{0, 1, \ldots, b/k - 1\}$ such that

$$\gamma^{\ell_s(k-1)q^e - \ell_t(k-1)q^h} = 1,$$

which implies that $\ell_s(k-1)q^e = \ell_t(k-1)q^h \pmod{q^{b/k} - 1}$. Therefore, there exists an integer $t$ such that

$$\ell_s(k-1)q^e = \ell_t(k-1)q^h + t(q^{b/k} - 1)$$

$$\ell_s q^e \cdot \frac{k - 1}{g} = \ell_t q^h \cdot \frac{k - 1}{g} + t \cdot \frac{q^{b/k} - 1}{g},$$

and thus,

$$\ell_s q^e \cdot \frac{k - 1}{g} = \ell_t q^h \cdot \frac{k - 1}{g} \left( \mod \frac{q^{b/k} - 1}{g} \right). \quad (9.10)$$

Since clearly, $\gcd\left(\frac{k - 1}{g}, \frac{q^{b/k} - 1}{g}\right) = 1$, it follows that $\frac{k - 1}{g}$ is invertible modulo $\frac{q^{b/k} - 1}{g}$. Therefore, (9.10) implies that $\ell_s q^e = \ell_t q^h (\mod \frac{q^{b/k} - 1}{g})$. Since $\gcd(q, \frac{q^{b/k} - 1}{g}) = 1$, it follows that $q$ is invertible modulo $\frac{q^{b/k} - 1}{g}$. Hence, we have that $\ell_s = \ell_t q^{h - e (\mod \frac{q^{b/k} - 1}{g})}$ if $h \geq e$ and $\ell_s q^{e - h} = \ell_t (\mod \frac{q^{b/k} - 1}{g})$ if $h < e$. Either way, it follows that $\ell_t$ and $\ell_s$, which are notations for $i_{j_1}$ and $i_{j_2}$, respectively, are in the same $q$-cyclotomic coset modulo $\frac{q^{b/k} - 1}{g}$, a contradiction to the choice of $i_1, \ldots, i_n$. Therefore, $1 \notin \Delta$, which implies that (9.9) is solvable, and the data collector may obtain $[Q]_{s,t}$ and $[W]_{s,t}$ for all distinct $s$ and $t$ in $[k]$.

Having this information, the data collector may consider the $i$-th block-
row of $Q$, excluding the diagonal element,

$\Phi_1 S_2 \left( \Phi_1^\top \cdots \Phi_{i-1}^\top \Phi_i^\top \cdots \Phi_k^\top \right),$

in which the matrix on the right is invertible by construction, and by Lemma 129. Hence, the data collector obtains $\Phi_1 S_2, \ldots, \Phi_k S_2$, out of which any $k - 1$ may once again be used to extract $S_2$ by the same argument. Clearly, $S_1$ may be obtained similarly from the submatrices $[W]_{s,t}$. ■

Note that in the above code $B = \frac{b^2(k-1)}{k} \left( 1 - \frac{1}{k} + \frac{1}{b} \right)$, and $\alpha k = \frac{b^2(k-1)}{k}$. Thus, the construction in this section does not provide an MSR code. However, $\frac{B}{\alpha k} \xrightarrow{k \to \infty} 1$, and thus the cut-set bound is achieved asymptotically. A detailed comparison with PM-MSR codes and numerical examples appear in the Subsection 9.A.5.2.

This construction can be used to obtain NMSR codes for $d > 2k - 2$ in a recursive manner. By following a very similar outline to that of [24, Th. 6], we have the following.

**Theorem 55** If there exists an $(n',k',d', B', q, \alpha, \beta)$ regenerating code $C'$ such that $\frac{B'}{\alpha k'} \xrightarrow{k' \to \infty} 1$, then there exists a $(n = n' - 1, k = k' - 1, d = d' - 1, B' = B - \alpha, q, \alpha, \beta)$ regenerating code $C$ such that $\frac{B}{\alpha k} \xrightarrow{k \to \infty} 1$.

**Proof.** Without loss of generality assume that $C'$ is systematic, and let $C$ be the code which results from **puncturing** the first systematic node of $C'$. It follows from the properties of $C'$ that $C$ is a code with $n = n' - 1$ nodes, in which any $d = d' - 1$ nodes can be used for repair, and any $k = k' - 1$ nodes may be used for reconstruction. Moreover, $B = B' - \alpha$, and

$$\frac{B}{\alpha k} = \frac{B' - \alpha}{\alpha(k' - 1)} = \frac{B'}{\alpha k'} \cdot \frac{k'}{k' - 1} - \frac{1}{k' - 1} \xrightarrow{k \to \infty} 1$$

■

Notice that in Theorem 55, if $d' = ik' + j$ then $d = ik + j + (i - 1)$. Hence, given the construction for $d = 2k - 2$, one may obtain NMSR codes for larger values of $d$. Moreover, it is evident that $\frac{B'}{\alpha k'} \leq 1$, $\frac{k'}{k' - 1} > 1$, and that $\frac{1}{k' - 1}$ is negligible as $k$ grows. Hence, the proof of Theorem 55 implies that $\frac{B}{\alpha k}$ tends to 1 faster than $\frac{B'}{\alpha k'}$ does.
9.A.5.2 The proximity of NMSR codes to MSR codes

In this subsection the construction from Subsection 9.A.5.1 is compared with PM-MSR codes for the case $d = 2k - 2$. A comparison for the case $d > 2k - 2$ will appear in future versions of this paper. Following the reasoning which is described in Subsection 9.A.4.1 the codes are compared over fields of even characteristic. That is, our codes are considered with $q = 2$, and since PM-MSR codes require $q \geq n(k - 1)$, they are considered with $q = 2^\lceil \log(n(k-1)) \rceil$.

Similar to Definition 49 and its subsequent discussion, EPM-MSR codes may also be defined. Note that a comparable loss of rate is apparent, not only due to the redundant representation, but also due to the symmetry which is required from the submatrices on the main diagonals of $S_1$ and $S_2$.

The codes PM-MSR and EPM-MSR are concatenated to themselves in order to obtain the same $n, k, d, \alpha, \beta$, and only then the resulting file size and code rate are compared. The comparison for general parameters appears in Table 9.3 in which the values of $\alpha, \beta$, and $B$ are given in bits. Note that as in Subsection 9.A.4.2 the value of $B$ for EPM-MSR is the number of information bits, rather than the number of bits in the redundant representation. Further, numerical examples are given in Table 9.4. Notice that it is possible to reduce the field size of PM-MSR codes in some cases [18]. Yet, we compare our NMSR codes to PM-MSR for simplicity and generality.

9.A.6 Connection to subspace codes

Several connections between MBR codes and subspace codes were observed in the past [22, 23, 6]. In these works, any storage node is associated with a subspace of dimension $\alpha$ in $F_q^B$, and stores a projection of $x$ on that subspace. However, these approaches do not outperform PM-MBR codes, and are only applicable for restricted sets of parameters. In this section it is shown that the results of Section 9.A.4.1 may also be attained by a variant of the subspace code approach. This connection may seem as a generalization of the techniques in Section 9.A.4.1 and besides being of independent interest, might be useful for applying our techniques in other scenarios. In what follows we present the essentials of subspace coding theory that are required for the construction which follows.

---

4That is, a multiplication of a spanning matrix of the subspace with the file $x$. 

311
Table 9.3: A comparison of parameters between our MSR codes (Subsection 9.A.5.1) and the PM-MSR codes [24, Sec. V] for general $n, k, d = 2k - 2$.

For any integer $m$, it is widely known that the vector space $\mathbb{F}_q^m$ may be endowed with a multiplication operation using the vector space isomorphism $\mathbb{F}_q^m \cong \mathbb{F}_{q^m}$. Hence, every subspace of $\mathbb{F}_q^m$ may be considered as a subspace of $\mathbb{F}_{q^m}$. In what follows, $\mathcal{G}_q(m, u)$ denotes the set of all $u$ dimensional subspaces of $\mathbb{F}_{q^m}$. A (constant dimension) subspace code is a subset of $\mathcal{G}_q(m, u)$, equipped with the subspace metric $d_S(U, V) = \dim U + \dim V - 2\dim(U \cap V)$.

Cyclic subspace codes are subspace codes which are closed under cyclic shifts, defined as follows. For a subspace $U \in \mathcal{G}_q(m, u)$, and a nonzero field element $\alpha \in \mathbb{F}_q^* \triangleq \mathbb{F}_{q^m} \setminus \{0\}$, the cyclic shift $\alpha U$ of $U$ is the set $\alpha U \triangleq \{\alpha \cdot u | u \in U\}$. According to the distributive law, for any $\alpha \in \mathbb{F}_q^*$, the set $\alpha U$ is a subspace of $\mathbb{F}_{q^m}$ of the same dimension as $U$.

A widely used notion in the theory of subspace codes is partial spreads (see for example, [4, Sec. III]). A constant dimension subspace code $\mathcal{C}$ in $\mathcal{G}_q(m, u)$ is called a partial spread if its minimum distance is $2u$, i.e., each two subspaces intersect trivially. Furthermore, if each vector in $\mathbb{F}_{q^m}$ is contained in some subspace in $\mathcal{C}$ then it is called a spread. It is known [4] that spreads exist if and only if $u$ divides $m$, and the set of all cyclic shifts of a subfield of $\mathbb{F}_{q^m}$ is a spread. In what follows, certain subsets of such spreads which satisfy an additional property are used. This property involves sets of
subspaces, in contrast with the distance property which only involves two, and is given by the following definitions. It is worth noting that partial spreads also appear in the work of [22].

**Definition 50** A set \( \{U_i\}_{i=1}^k \subseteq \mathcal{G}_q(m, u) \) is called an independent set if \( \dim \left( \sum_{i=1}^k U_i \right) = \sum_{i=1}^k \dim U_i \).

It is readily verified that a set \( \{U_i\}_{i=1}^k \) is an independent set if and only if for all \( j \in [k] \), we have that \( U_j \cap \left( \sum_{i \in [k]\setminus\{j\}} U_i \right) = \{0\} \). Further, if \( \{U_i\}_{i=1}^k \subseteq \mathcal{G}_q(b, u) \) is an independent set then \( u \cdot k \leq m \).

**Definition 51** A set \( \{U_i\}_{i=1}^n \subseteq \mathcal{G}_q(m, u) \) is called an every-\( k \) independent set if for every \( T \subseteq [n], |T| = k \) we have that \( \{U_i\}_{i \in T} \) is an independent set.

**Remark 15** For \( u = 1 \) and \( m = k \), an every-\( k \) independent set may easily be obtained by considering any \([n, k]\) MDS code over \( \mathbb{F}_q \), and taking as \( \{U_i\}_{i=1}^n \) the linear spans of the columns of its generator matrix [27, p. 119, Ex. 4.1]. Further, it can be shown (see Theorem 58 in the Appendix) that for an integer \( b \) which is a multiple of \( k \), if an every-\( k \) independent set
\( \{U_i\}_{i=1}^n \subseteq \mathcal{G}_q(b, b/k) \) exists, then there exists an \([n, k]\) MDS code over \( \mathbb{F}_{q^{b/k}} \) which is linear over \( \mathbb{F}_q \). Therefore, the notions of every-\( k \) independent sets are equivalent to linear MDS codes up to a certain extent.

The parameters \( q, n, k, d \) in this section are required to satisfy the conditions \( A1 \) and \( A2 \) from Section 9.A.4. In addition we also have that 
\[
B = \frac{b(b+1)}{2} + b^2 \left( \frac{d}{k} - 1 \right),
\]
where for convenience we denote \( c \triangleq \frac{db}{k} - b \), so that 
\[
B = \frac{b(b+1)}{2} + bc.
\]
Notice that \( c \) is chosen such that 
\[
bk = b + cd.
\]
The techniques in this section require to fix a certain mapping from \( \mathbb{F}_{q^{b+c}} \) to \( \mathbb{F}_{b+c}^d \), which is done as follows.

Since \( \frac{b}{k} = \frac{b+c}{d} \) it follows that \( d \) divides \( b + c \), and hence \( \mathbb{F}_{q^{b/k}} = \mathbb{F}_{q^{b+c}/d} \) is a subfield of \( \mathbb{F}_{q^{b+c}} \). Hence, let \( \mathcal{V} \triangleq \{v_1, \ldots, v_d\} \) be a basis of \( \mathbb{F}_{q^{b+c}} \) over \( \mathbb{F}_{q^{b+c}/d} \), and let \( \mathcal{U} \triangleq \{u_1, \ldots, u_{(b+c)/d}\} \) be a basis of \( \mathbb{F}_{q^{b+c}/d} \) over \( \mathbb{F}_q \). Since \( \mathcal{V} \) and \( \mathcal{U} \) are bases, it follows that
\[
\mathcal{VU} \triangleq \left\{ v_i \cdot u_j \mid 1 \leq i \leq d, 1 \leq j \leq \frac{b+c}{d} \right\}
\]
is a basis of \( \mathbb{F}_{q^{b+c}} \) over \( \mathbb{F}_q \) (see Lemma 137 in the appendix). The basis \( \mathcal{VU} \) is used to fix a mapping \( \Phi \) of the elements of \( \mathbb{F}_{q^{b+c}} \) to the elements of \( \mathbb{F}_q^{b+c} \). Identify the indices \( 1, \ldots, b + c \) with
\[
(1, 1), (1, 2), \ldots, (1, \frac{b+c}{d}), (2, 1), (2, 2), \ldots, (d, \frac{b+c}{d}),
\]
respectively, and for \( w \in \mathbb{F}_{q^{b+c}} \), \( w = \sum_{i,j} w_{i,j} v_i u_j \), where \( w_{i,j} \in \mathbb{F}_q \) for all \( 1 \leq i \leq d, 1 \leq j \leq \frac{b+c}{d} \), define \( \Phi(w) \) as the vector of length \( b + c \) over \( \mathbb{F}_q \) which contains \( w_{i,j} \) in entry \((i, j)\). Notice that the function \( \Phi \) is linear over \( \mathbb{F}_q \), i.e., every \( w, w' \in \mathbb{F}_{q^{b+c}} \) and every \( \lambda, \mu \in \mathbb{F}_q \) satisfy \( \Phi(\lambda w + \mu w') = \lambda \Phi(w) + \mu \Phi(w') \). To avoid cumbersome notation, we use \( w \) instead of \( \Phi(w) \) wherever it is clear from context.

This section is organized as follows. In Subsection 9.A.6.1 we construct an every-\( d \) independent set, for whom an additional property is proved. This property is a consequence of the particular representation \( \Phi \) of \( \mathbb{F}_{q^{b+c}} \) as vectors in \( \mathbb{F}_q^{b+c} \). This set is then used to construct NMBR codes for any \( d \geq k \) over any field \( \mathbb{F}_q \) in Subsection 9.A.6.2. The encoding matrix of this code is shown to possess a Vandermonde-like structure in subsection 9.A.6.3.
9.A.6.1 Construction of an every-$d$ independent set in $G_q(b+c, b+c/d)$

Since $b \geq \log_q n \cdot k$, it follows that $q^{b/k} = q^{(b+c)/d} \geq n$, and hence there exists an independent set over $\mathbb{F}_{q^{b+c/d}}$, with $\gamma_i \neq \gamma_j$ for all distinct $i$ and $j$ in $[n]$, which implies that every $d \times d$ submatrix of $A'$ and every $k \times k$ submatrix of $A$ are invertible.

Using the basis $V = \{v_1, \ldots, v_d\}$ which was mentioned above, for every $i \in [n]$ define $\beta_i = \sum_{j=1}^d \gamma_{j-1} v_j$, and $\beta'_i = \sum_{j=1}^k \gamma_{j-1} v_j$. The proof of the following lemma relies on the fact that any $d$ (resp. $k$) columns of $A'$ (resp. $A$) are linearly independent over $\mathbb{F}_{q^{b+c/d}}$.

**Lemma 134**

A. Every $k$ distinct elements in $\{\beta'_i\}_{i=1}^n$ are linearly independent over $\mathbb{F}_{q^{b+c/d}}$.

B. Every $d$ distinct elements in $\{\beta_i\}_{i=1}^n$ are linearly independent over $\mathbb{F}_{q^{b+c/d}}$.

**Proof.** Since the set $\{v_1, \ldots, v_d\}$ is a basis of $\mathbb{F}_{q^{b+c}}$ over $\mathbb{F}_{q^{(b+c)/d}}$, it follows that the set $\{v_1, \ldots, v_k\}$ is an independent set over $\mathbb{F}_{q^{(b+c)/d}}$. Hence, the proofs of Part A and Part B are similar, and we list below only the proof of Part B.

Let $J = \{j_1, \ldots, j_d\}$ be a subset of $[n]$ of size $d$, and assume that $\sum_{i=1}^d d_i \beta_{j_i} = 0$ for some coefficients $d_i \in \mathbb{F}_{q^{(b+c)/d}}$. According to the def-
inition of the $\beta_i$-s we have that

$$\sum_{i=1}^d d_i \cdot \beta_{j_i} = \sum_{i=1}^d d_i \sum_{t=1}^d \gamma_{j_i}^{t-1} v_t = \sum_{i=1}^d \sum_{t=1}^d d_i \gamma_{j_i}^{t-1} v_t = \sum_{t=1}^d \sum_{i=1}^d d_i \gamma_{j_i}^{t-1} v_t.$$

Since $d_i$ and $\gamma_{j_i}^{t-1}$ are elements of $\mathbb{F}_{q^{(b+c)/d}}$ for all $i$ and $t$, we have that (9.13) is a linear combination of the independent set $\{v_{i,1}\}_{i=1}^d$ over $\mathbb{F}_{q^{(b+c)/d}}$. Therefore, $\sum_{i=1}^d d_i \gamma_{j_i}^{t-1} = 0$ for all $t \in [d]$, which implies that $\sum_{i=1}^d d_i c_{j_i} = 0$, where $c_{j_i}$ is the $i$-th column of $A'$. Since $A'$ is a Vandermonde matrix, the column vectors $\{c_{j_i}\}_{i=1}^d$ are independent over $\mathbb{F}_{q^{(b+c)/d}}$, and thus $d_i = 0$ for all $i$. ■

For $i \in [n]$ let $V_i \triangleq \beta_i \mathbb{F}_{q^{(b+c)/d}}$ and $V'_i \triangleq \beta'_i \mathbb{F}_{q^{(b+c)/d}}$, and notice that $V_i, V'_i \in G_q (b + c, \frac{b+c}{d})$.

**Lemma 135**

A. The set $\{V'_i\}_{i=1}^n$ is an every-$k$ independent set.

B. The set $\{V_i\}_{i=1}^n$ is an every-$d$ independent set.

**Proof.** To prove Part A let $J = \{j_1, \ldots, j_k\}$ be a subset of $[n]$ of size $k$. Consider the function $f : \mathbb{F}_{q^{b/k}}^k \to \sum_{i=1}^k V'_{j_i}$ which maps $(\ell_1, \ldots, \ell_k)$ to $\sum_{i=1}^k \beta'_{j_i} \ell_i$. It is readily verified that $\dim \left( \sum_{i=1}^k V'_{j_i} \right) = b$ if and only if $f$ is injective. Assume for contradiction that $f$ is not injective, i.e., there exist two different tuples of elements $(e_1, \ldots, e_k) \in \mathbb{F}_{q^{b/k}}^k$ and $(f_1, \ldots, f_k) \in \mathbb{F}_{q^{b/k}}^k$ such that

$$\sum_{i=1}^k \beta'_{j_i} e_i = \sum_{i=1}^k \beta'_{j_i} f_i.$$

Therefore, $\sum_{i=1}^k \beta'_{j_i} (e_i - f_i) = 0$, which by Lemma 134 Part A implies that $e_i = f_i$ for all $i$, a contradiction. This implies that $\dim \left( \sum_{i=1}^k V'_{j_i} \right) = b = k \cdot \frac{b}{k} = \sum_{i=1}^k \dim V'_{j_i}$, as required.

To prove Part B let $T = \{t_1, \ldots, t_d\}$ be a subset of $[n]$ of size $d$. Consider the function $g : \mathbb{F}_{q^{(b+c)/d}}^d \to \sum_{i=1}^d V_{t_i}$ which maps $(\ell_1, \ldots, \ell_d)$ to $\sum_{i=1}^d \beta_{j_i} \ell_i$.
It is readily verified that \( \text{dim} \left( \sum_{i=1}^{k} V_{j_i} \right) = b + c \) if and only if \( g \) is injective. Assume for contradiction that \( g \) is not injective, i.e., there exist two different tuples of elements \((e_1, \ldots, e_d) \in \mathbb{F}_q^{(b+c)/d}\) and \((f_1, \ldots, f_d) \in \mathbb{F}_q^{(b+c)/d}\) such that

\[
\sum_{i=1}^{d} \beta_{ji} e_i = \sum_{i=1}^{d} \beta_{ji} f_i.
\]

Therefore, \( \sum_{i=1}^{d} \beta_{ji} (e_i - f_i) = 0 \), which by Lemma \ref{lem:tech_lem} Part B implies that \( e_i = f_i \) for all \( i \), a contradiction. Hence, \( \text{dim} \left( \sum_{i=1}^{k} V_{j_i} \right) = b + c = d \cdot \frac{b+c}{d} = \sum_{i=1}^{d} \text{dim} V_{j_i} \).

The following technical lemma is essential for the construction which follows. The proof of this lemma uses the function \( \Phi \), which maps an element of \( \mathbb{F}_q^{b+c} \) to its representation according to the basis \( \mathcal{U} \) (9.11), as defined in the beginning of Section 9.A.6.

**Lemma 136** For all \( i \in [n] \), there exists a matrix \( \mathcal{M}_i \in \mathbb{F}_q^{(b+c)/d \times b} \) such that \( V_i = \langle \mathcal{M}_i | N_i \rangle \) for some matrix \( N_i \in \mathbb{F}_q^{(b+c)/d \times c} \) and \( V'_i = \langle \mathcal{M}_i | \mathbf{0} \rangle \), where \( \mathbf{0} \) is the \( \frac{b+c}{d} \times c \) zero matrix.

**Proof.** By definition, the subspace \( V_i \) is spanned by the vectors \( \{ \beta_i u_j \}^{(b+c)/d} = \{ \sum_{t=1}^{d} v_t \gamma_i^{t-1} u_j \}^{(b+c)/d} \), and similarly, the subspace \( V'_i \) is spanned by the vectors \( \{ \beta'_i u_j \}^{(b+c)/d} = \{ \sum_{t=1}^{k} v_t \gamma_i^{t-1} u_j \}^{(b+c)/d} \).

Since the function \( \Phi \) is linear, it follows that for every \( j \in [\frac{b+c}{d}] \) and every \( i \in [n] \), we have that

\[
\Phi \left( \sum_{t=1}^{d} v_t \gamma_i^{t-1} u_j \right) = \sum_{t=1}^{d} \Phi \left( v_t \gamma_i^{t-1} u_j \right).
\]

Moreover, it follows from the definition of \( \Phi \) that for all \( t \in [d] \), the vector \( \Phi \left( v_t \gamma_i^{t-1} u_j \right) \) contains the representation of \( \gamma_i^{t-1} u_j \in \mathbb{F}_q^{(b+c)/d} \) by the basis \( \mathcal{U} \) in entries \( (t, 1), \ldots, (t, \frac{b+c}{d}) \), and zero elsewhere. That is, if \( \Psi : \mathbb{F}_q^{(b+c)/d} \rightarrow \mathbb{F}_q^{(b+c)/d} \) is the function which maps an element of \( \mathbb{F}_q^{(b+c)/d} \) to its
representation by \( U \), and \( \Psi(\gamma_i^{t-1}u_j) \triangleq (\eta_1, \ldots, \eta_{(b+c)/d}) \), then

\[
\Phi(v_t\gamma_i^{t-1}u_j) = \Phi \left( v_t \sum_{s=1}^{(b+c)/d} \eta_s u_s \right)
= \Phi \left( \sum_{s=1}^{(b+c)/d} \eta_s v_t u_s \right)
= (0, \ldots, 0, \eta_1, \ldots, \eta_{(b+c)/d}, 0, \ldots, 0).
\]

Hence, for every \( j \in \left[ \frac{b+c}{d} \right] \) and every \( i \in [n] \) we have that

\[
\Phi(\beta_i u_j) = \Phi \left( \sum_{t=1}^{d} v_t\gamma_i^{t-1}u_j \right) = \Psi(u_j) \circ \Psi(\gamma_i u_j) \circ \cdots \circ \Psi(\gamma_{d-1}^{i-1}u_j),
\]

where \( \circ \) denotes concatenation, and similarly,

\[
\Phi(\beta'_i u_j) = \Phi \left( \sum_{t=1}^{k} v_t\gamma_i^{t-1}u_j \right) = \Psi(u_j) \circ \Psi(\gamma_i u_j) \circ \cdots \circ \Psi(\gamma_{k-1}^{i-1}u_j) \circ \Psi(0) \circ \cdots \circ \Psi(0) .
\]

Hence, for every \( i \in [n] \), the subspaces \( V_i \) and \( V'_i \) have the spanning matrices

\[
V_i = \left\langle \begin{pmatrix}
\Psi(u_1) & \Psi(\gamma_i u_1) & \cdots & \Psi(\gamma_{d-1}^{i-1} u_1) \\
\Psi(u_2) & \Psi(\gamma_i u_2) & \cdots & \Psi(\gamma_{d-1}^{i-1} u_2) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_i u_{(b+c)/d}) & \cdots & \Psi(\gamma_{d-1}^{i-1} u_{(b+c)/d})
\end{pmatrix} \right\rangle,
\]

and

\[
V'_i = \left\langle \begin{pmatrix}
\Psi(u_1) & \Psi(\gamma_i u_1) & \cdots & \Psi(\gamma_{k-1}^{i-1} u_1) & 0 & \cdots & 0 \\
\Psi(u_2) & \Psi(\gamma_i u_2) & \cdots & \Psi(\gamma_{k-1}^{i-1} u_2) & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_i u_{(b+c)/d}) & \cdots & \Psi(\gamma_{k-1}^{i-1} u_{(b+c)/d}) & 0 & \cdots & 0
\end{pmatrix} \right\rangle.
\]

\[\blacksquare\]
Thus, for any given set of \( n \) distinct elements \( \{ \gamma_i \}_{i=1}^n \) in \( \mathbb{F}_{q^{(b+c)/d}} \), define the following matrices which will be used in the sequel.

\[
\mathcal{M}_i \triangleq \begin{pmatrix}
\Psi(u_1) & \Psi(\gamma_i u_1) & \cdots & \Psi(\gamma_i^{k-1} u_1) \\
\Psi(u_2) & \Psi(\gamma_i u_2) & \cdots & \Psi(\gamma_i^{k-1} u_2) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_i u_{(b+c)/d}) & \cdots & \Psi(\gamma_i^{k-1} u_{(b+c)/d})
\end{pmatrix} \in \mathbb{F}_q^{b \times b}
\]

(9.14)

\[
\mathcal{N}_i \triangleq \begin{pmatrix}
\Psi(\gamma_i^k u_1) & \cdots & \Psi(\gamma_i^{d-1} u_1) \\
\Psi(\gamma_i^k u_2) & \cdots & \Psi(\gamma_i^{d-1} u_2) \\
\vdots & \ddots & \vdots \\
\Psi(\gamma_i^k u_{(b+c)/d}) & \cdots & \Psi(\gamma_i^{d-1} u_{(b+c)/d})
\end{pmatrix} \in \mathbb{F}_q^{b \times c}
\]

(9.15)

**9.A.6.2 Nearly MBR codes from subspace codes**

Arrange the symbols of the file \( x \in \mathbb{F}_q^B \) as in (9.3). Encode \( x \) as follows,

\[
\begin{pmatrix}
\mathcal{M}_1 & \mathcal{N}_1 \\
\mathcal{M}_2 & \mathcal{N}_2 \\
\vdots & \vdots \\
\mathcal{M}_n & \mathcal{N}_n
\end{pmatrix}
\begin{pmatrix}
S & T \\
T^\top & 0
\end{pmatrix}
= 
\begin{pmatrix}
\mathcal{M}_1 S + \mathcal{N}_1 T^\top & \mathcal{M}_1 T \\
\mathcal{M}_2 S + \mathcal{N}_2 T^\top & \mathcal{M}_2 T \\
\vdots & \vdots \\
\mathcal{M}_n S + \mathcal{N}_n T^\top & \mathcal{M}_n T
\end{pmatrix},
\]

where \( \mathcal{M}_i \in \mathbb{F}_q^{b/k \times b} \) and \( \mathcal{N}_i \in \mathbb{F}_q^{b/k \times c} \) were defined in (9.14) and (9.15), respectively. By the definition of the matrices \( \mathcal{M}_i, \mathcal{N}_i, S, \) and \( T \), we have that \( \alpha = \frac{(b+c)^2}{d} = \frac{b^2}{k^2} \cdot d \).

**Theorem 56** In the above code, exact repair of any failed node may be achieved by downloading \( \beta = \frac{(b+c)^2}{d^2} = \frac{b^2}{k^2} \cdot d \) field elements from any \( d \) of the remaining nodes.

**Proof.** Assume that node \( i \) failed, and \( D = \{ j_1, \ldots, j_d \} \) is a subset of \( [n] \) of size \( d \) such that \( i \notin D \). To repair node \( i \), every node \( j \in J \) computes \( (\mathcal{M}_j \mid \mathcal{N}_j) X(\mathcal{M}_i \mid \mathcal{N}_i)^\top \), which is a \( \frac{b+c}{d} \times \frac{b+c}{d} \) matrix over \( \mathbb{F}_q \), and sends it to
the newcomer. The newcomer obtains
\[
\begin{pmatrix}
(M_{j_1}|N_{j_1})X(M_{i}|N_{i})^T \\
(M_{j_2}|N_{j_2})X(M_{i}|N_{i})^T \\
\vdots \\
(M_{j_d}|N_{j_d})X(M_{i}|N_{i})^T
\end{pmatrix}
= \begin{pmatrix}
(M_{j_1}|N_{j_1}) \\
(M_{j_2}|N_{j_2}) \\
\vdots \\
(M_{j_d}|N_{j_d})
\end{pmatrix}
\cdot X 
\cdot \begin{pmatrix}
(M_{i}|N_{i})^T
\end{pmatrix}
= \begin{pmatrix}
(M_{j_1}|N_{j_1}) \\
(M_{j_2}|N_{j_2}) \\
\vdots \\
(M_{j_d}|N_{j_d})
\end{pmatrix}^\top
\cdot (M|N)_D X (M|N)_i^\top.
\]

According to Lemma 136 we have that $\langle (M_{j_i}|N_{j_i}) \rangle = V'_{j_i}$, and by Lemma 135, Part B, we have that the row span of $(M|N)_D \in \mathbb{F}_q^{(b+c)\times(b+c)}$ is $\mathbb{F}_q^{b+c}$, and hence it is invertible. Thus, the newcomer may obtain the missing data by multiplying by $(M|N)^{-1}D$ and transposing.

**Theorem 57** In the above code, reconstruction may be achieved by downloading $\alpha = \frac{(b+c)^2}{d}$ field elements from any $k$ nodes.

**Proof.** Let $K = \{j_1, \ldots, j_k\}$ be a subset of $[n]$ of size $k$. By downloading the entire content $(M_{j_i}|N_{j_i})X$ from node $j_i$ for each $j_i \in J$, the data collector obtains
\[
\begin{pmatrix}
M_{j_1}S + N_{j_1}T^\top & M_{j_1}T \\
M_{j_2}S + N_{j_2}T^\top & M_{j_2}T \\
\vdots \\
M_{j_k}S + N_{j_k}T^\top & M_{j_k}T
\end{pmatrix}
\equiv
\begin{pmatrix}
M_{j_1}S + N_{j_1}T^\top \\
M_{j_2}S + N_{j_2}T^\top \\
\vdots \\
M_{j_k}S + N_{j_k}T^\top
\end{pmatrix}
\cdot M_K T.
\]

Recall that according to Lemma 136 we have that $\langle (M_{j_i}|0) \rangle = V'_{j_i}$. Since $\{V'_{j_i}\}_{i=1}^k$ is an independent set by Lemma 135, Part A, it follows that $\dim(\sum_{i=1}^k V'_{j_i}) = \sum_{i=1}^k \dim(V'_{j_i}) = \frac{b}{k} \cdot k = b$, and hence,
\[
\text{rank}
\begin{pmatrix}
M_{j_1} & 0 \\
M_{j_1} & 0 \\
\vdots & \vdots \\
M_{j_k} & 0
\end{pmatrix}
= b
\]
which implies that $M_K \in \mathbb{F}_q^{b\times b}$ is invertible. Therefore, the data collector may extract the rightmost $c$ columns of his data, multiply by $M_K^{-1}$, and obtain $T$. Having $T$, the data collector may similarly obtain $S$ as well.

Since $\beta = \frac{(b+c)^2}{d^2}$, $\alpha = \frac{(b+c)^2}{d^2}$, we have that $\beta d = \alpha$, and hence this code enables minimum bandwidth repair. Notice also that rate $\frac{B}{\alpha n}$ equals
\[
\frac{k^2}{dn} \left( \frac{1}{2b} + \frac{d}{k} \frac{1}{2} \right) \approx \frac{2k-k^2/d}{2n}.
\]
Hence, this construction is identical in parameters to the one given in Section 9.A.4.1 and the analysis in Section 9.A.4.2 applies to it as well. Yet, the subspace interpretation which is given in this section could possibly be implemented with alternative constructions of every-\(d\) independent sets, which may be of independent mathematical interest.

9.A.6.3 Vandermonde matrix structure

The repair algorithm in the proof of Theorem 56 relies on inverting a matrix of the form \((M|N)_D\) for some \(d\)-subset \(D\) of \([n]\), and the reconstruction algorithm in the proof of Theorem 57 relies on inverting a matrix of the form \(M_K\) for some \(k\)-subset \(K\) of \([n]\). These matrix posses a hidden Vandermonde structure which may be utilized for the inversion process by either an explicit formula \[12, \text{Sec. 1.2.3, Ex. 40}\] or a Reed-Solomon decoder \[27, \text{Ch. 6}\]. The Vandermonde structure of \(M_K\) is presented, and the one of \((M|N)_D\) may be obtained similarly.

According to (9.14), for \(K = \{j_1, \ldots, j_k\}\) the \(b \times b\) matrix \(M_K\) over \(\mathbb{F}_q\) is of the form

\[
M_K = \begin{pmatrix}
\Psi(u_1) & \Psi(\gamma_{j_1} u_1) & \cdots & \Psi(\gamma_{j_1}^{k-1} u_1) \\
\Psi(u_2) & \Psi(\gamma_{j_1} u_2) & \cdots & \Psi(\gamma_{j_1}^{k-1} u_2) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_{j_1} u_{(b+c)/d}) & \cdots & \Psi(\gamma_{j_1}^{k-1} u_{(b+c)/d}) \\
\Psi(u_1) & \Psi(\gamma_{j_2} u_1) & \cdots & \Psi(\gamma_{j_2}^{k-1} u_1) \\
\Psi(u_2) & \Psi(\gamma_{j_2} u_2) & \cdots & \Psi(\gamma_{j_2}^{k-1} u_2) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_{j_2} u_{(b+c)/d}) & \cdots & \Psi(\gamma_{j_2}^{k-1} u_{(b+c)/d}) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_1) & \Psi(\gamma_{j_k} u_1) & \cdots & \Psi(\gamma_{j_k}^{k-1} u_1) \\
\Psi(u_2) & \Psi(\gamma_{j_k} u_2) & \cdots & \Psi(\gamma_{j_k}^{k-1} u_2) \\
\vdots & \vdots & \ddots & \vdots \\
\Psi(u_{(b+c)/d}) & \Psi(\gamma_{j_k} u_{(b+c)/d}) & \cdots & \Psi(\gamma_{j_k}^{k-1} u_{(b+c)/d})
\end{pmatrix}, \quad (9.16)
\]

where the function \(\Psi\) maps an element in \(\mathbb{F}_{q^{k/k}}\) to its vector representation.
in $\mathbb{F}_q^{b/k}$ using the basis $\mathcal{U} = \{u_1, \ldots, u_{b/k}\}$. Hence, by inverting $\Psi$, the matrix $\mathcal{M}_K$, which is a $b \times b$ matrix over $\mathbb{F}_q$, can be considered as a $b \times k$ matrix over $\mathbb{F}_{q^{b/k}}$. In addition, for $i \in \left[\frac{b}{k}\right]$, if $M_{J_i}$ is the $k \times k$ matrix over $\mathbb{F}_{q^{b/k}}$ which consists of all the $i$-th rows from each row-block in (9.16), then

$$M_{J_i} = u_i \cdot \begin{pmatrix} 1 & \gamma_{j_1} & \gamma_{j_1}^2 & \cdots & \gamma_{j_1}^{k-1} \\ 1 & \gamma_{j_2} & \gamma_{j_2}^2 & \cdots & \gamma_{j_2}^{k-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \gamma_{j_k} & \gamma_{j_k}^2 & \cdots & \gamma_{j_k}^{k-1} \end{pmatrix}.$$ 

Therefore, the matrix $\mathcal{M}_K$ can be seen as a row-interleaving of $\frac{b}{k}$ constant multiples of a Vandermonde matrix over $\mathbb{F}_{q^{b/k}}$.

### 9.A.7 Discussion and future research

In this paper, asymptotically optimal regenerating codes were introduced. These codes attain the cut-set bound asymptotically as the reconstruction degree $k$ increases, and may be defined over any field if the file size is reasonably large. Further, these codes enjoy several properties which are inherited from product matrix codes, such as the fact that helper nodes do not need to know the identity of each other\(^5\), and the ability to add an extra storage node without encoding the file anew.

It is evident from Table 9.2 and Table 9.4 that for $q = 2$, a small loss of code rate is apparent already for feasible values of $k$, and clearly, similar results hold for larger $q$ as well. Since large finite field arithmetics is often infeasible, our results contribute to the feasibility of storage codes.

The research of storage codes gained a considerable amount of attention lately. In particular, the results of [24], which inspired ours, was expanded and improved in few recent papers. For example, [3] generalized the PM-MBR construction to achieve other points of the trade-off through minor matrices, and [19] presented an MBR code which supports an arbitrary number of helper nodes in the repair process. Among the research directions we currently pursue are the application of the techniques from the current paper to the aforementioned works, as well as to high rate MSR constructions, and

---

\(^5\)Although this property is apparent in most regenerating codes construction, some constructions do require otherwise, such as [3], and some of the work of [25].
analyzing the encoding, decoding, repair, and reconstruction complexities of our codes in comparison with PM codes.

Acknowledgments

The work of Netanel Raviv was supported in part by the Israeli Science Foundation (ISF), Jerusalem, Israel, under Grant no. 10/12, The IBM Ph.D. fellowship, and the Mitacs organization, under the Globalink Israel-Canada Innovation Initiative. The author would like to express his sincere gratitude to Prof. Frank Kschischang, Prof. Tuvi Etzion, and Prof. Itzhak Tamo for many insightful discussions.
Bibliography


325


Appendix A - Systematic encoding

Similar to [24, Sec. IV.B], the codes presented in Section 9.A.4 and Section 9.A.6 have a systematic form. This form can be directly obtained by using a Cauchy matrix, without the need to apply an invertible linear transform on the encoding matrix. However, using this Cauchy matrix requires a modification of Condition A1 as explained in the sequel.
Recall that the encoding matrix \( M \) has a block-Vandermonde structure. Alternatively, it can be obtained from a \( k \times n \) Vandermonde matrix over \( \mathbb{F}_{q^b/k} \) and applying the representation of \( \mathbb{F}_{q^b/k} \) as powers of a companion matrix (Lemma 127). The essential property of the matrix \( M \) which is used throughout the paper is the fact that any \( d \times d \) matrix, and every \( k \times k \) submatrix which consists of rows 1 through \( k \) is invertible. Hence, we may alternatively consider

\[
A'' \triangleq \begin{pmatrix} I_k & E \\ 0 & F \end{pmatrix},
\]

where \((E)\) is a \( d \times (n-k) \) Cauchy matrix over \( \mathbb{F}_{q^b/k} \), and \( I_k \) is the \( k \times k \) identity matrix. Since every square submatrix of a Cauchy matrix is invertible, applying the matrix representation to \( A'' \) will result in a systematic version of the code. However, to have a \( d \times (n-k) \) Cauchy matrix over \( \mathbb{F}_{q^b/k} \) one must update Condition A1 to \( b \geq \log_q(n+d-k) \cdot k \).

Similarly, in Section 9.A.6 we have that the crucial property of the matrix \( A' \) (resp., the matrix \( A \)) in (9.12) is the fact that any set of \( d \) (resp. \( k \)) columns in it are linearly independent. Hence, a nearly equivalent code may be obtained by considering the above matrix \( A'' \in \mathbb{F}_{q^b/k}^{d \times n} \). Further, the encoding matrices \( M_i \) and \( N_i \), given in (9.14) and (9.15), are replaced by

\[
\mathcal{M}_i \triangleq \begin{pmatrix} \Psi(a_{1,i} u_1) & \Psi(a_{2,i} u_1) & \cdots & \Psi(a_{k,i} u_1) \\ \Psi(a_{1,i} u_2) & \Psi(a_{2,i} u_2) & \cdots & \Psi(a_{k,i} u_2) \\ \vdots & \vdots & \ddots & \vdots \\ \Psi(a_{1,i} u_{(b+c)/d}) & \Psi(a_{2,i} u_{(b+c)/d}) & \cdots & \Psi(a_{k,i} u_{(b+c)/d}) \end{pmatrix}
\]

\[
\mathcal{N}_i \triangleq \begin{pmatrix} \Psi(a_{k+1,i} u_1) & \cdots & \Psi(a_{d,i} u_1) \\ \Psi(a_{k+1,i} u_2) & \cdots & \Psi(a_{d,i} u_2) \\ \vdots & \ddots & \vdots \\ \Psi(a_{k+1,i} u_{(b+c)/d}) & \cdots & \Psi(a_{d,i} u_{(b+c)/d}) \end{pmatrix},
\]
where \( a_{i,j} \) denotes the \((i,j)\)-th entry of \( A'' \). Consequently, we have that
\[
\begin{pmatrix}
M_1 \\
M_2 \\
\vdots \\
M_k
\end{pmatrix} = I_b
\]
and \( N_1 = N_2 = \ldots = N_k = 0 \), which implies that nodes 1 through \( k \) together hold the information \((S\ T)\). Hence, by using \( A'' \) instead of \( A' \), a systematic version of the code is obtained. Notice that Condition \([A1]\) is to be replaced by \( b \geq \log_q(n + d - k) \cdot k \) as well.

Appendix B - Omitted proofs

**Theorem 58** For any integers \( b \) and \( k \) such that \( b|k \) and for any prime power \( q \), if there exists an every-\( k \) independent set \( \{U_i\}_{i=1}^n \subseteq G_q(b,\frac{b}{k}) \), then there exists an \([n,k]\) MDS code over \( \mathbb{F}_{q^{b/k}} \) which is linear over \( \mathbb{F}_q \).

**Proof.** The required MDS code is defined by a function \( f : \mathbb{F}_{q^{b/k}}^k \to \mathbb{F}_{q^{b/k}}^n \) such that for any \( c \in \text{Im} f \), where \( \text{Im} f \) is the image of \( f \), the preimage \( f^{-1}(c) \) can be computed given any \( k \) symbols of \( c \). For each \( i \in [n] \) let \( M_i \in \mathbb{F}_q^{(b/k)\times b} \) be a matrix such that \( \langle M_i \rangle = U_i \), fix any basis \( U \equiv \{u_1,\ldots,u_{b/k}\} \) of \( \mathbb{F}_{q^{b/k}} \) over \( \mathbb{F}_q \), and use \( U \) to consider a given \( x \in \mathbb{F}_{q^{b/k}}^k \) as a vector in \( \mathbb{F}_{q^b} \). For each \( j \in [n] \) define \( f(x)_j \) as \( x M_j^\top \), where \( U \) is once again used to represent a vector in \( \mathbb{F}_{q^{b/k}}^b \) as an element in \( \mathbb{F}_{q^{b/k}}^b \), that is, \( f(x)_j = \sum_{i=1}^{b/k} (x M_j^\top)_i u_i \).

To prove that \( \text{Im} f \) is an MDS code, let \( J \equiv \{j_1,\ldots,j_k\} \) be a subset of \([n]\) of size \( k \), and assume that \( c_{j_1},\ldots,c_{j_k} \) are given for some \( c \in \text{Im} f \). Since \( \sum_{i=1}^{k} U_{j_i} = \mathbb{F}_q^b \), and since \( c_{j_i} = x M_{j_i}^\top \) for some \( x \in \mathbb{F}_q^b \) and for all \( i \in [k] \), it follows that the matrix \( \begin{pmatrix} M_{j_1}^\top & M_{j_2}^\top & \cdots & M_{j_k}^\top \end{pmatrix} \) is invertible, and thus \( x \) can be computed from \( c_{j_1},\ldots,c_{j_k} \).

To prove that \( \text{Im} f \) is linear over \( \mathbb{F}_q \), for \( x = (x_1,\ldots,x_k) \in \mathbb{F}_{q^{b/k}}^k \) denote its representation in \( \mathbb{F}_{q^b}^b \) as \( x = (x_{1,1},x_{1,2},\ldots,x_{1,b/k},x_{2,1},\ldots,x_{k,b/k}) \). For all
\( x, y \in \mathbb{F}_{q^{b/k}} \) and \( j \in [n] \) we have that

\[
f(x + y)_j = \sum_{i=1}^{b/k} ((x + y)M_j^T)_iu_i = \sum_{i=1}^{b/k} (xM_j^T)_iu_i + \sum_{i=1}^{b/k} (yM_j^T)_iu_i = f(x)_j + f(y)_j,
\]

and hence \( f(x + y) = f(x) + f(y) \). Since for \( a \in \mathbb{F}_q \) we have that \( f(ax)_j = \sum_{i=1}^{b/k} (axM_j^T)_iu_i = af(x) \), the claim follows.

**Lemma 137** The set \( \mathcal{U} \) from (9.11) is a basis of \( \mathbb{F}_{q^{b+c}} \) over \( \mathbb{F}_q \).

**Proof.** Assume that

\[
\sum_{i=1}^{d} \sum_{j=1}^{b+c/d} w_{i,j}v_iu_j = 0
\]

for some \( w_{i,j} \) in \( \mathbb{F}_q \), and notice that

\[
\sum_{i=1}^{d} \sum_{j=1}^{b+c/d} w_{i,j}v_iu_j = \sum_{i=1}^{d} v_i \sum_{j=1}^{b+c/d} w_{i,j}u_j = 0. \tag{9.17}
\]

Since the elements of the basis \( \mathcal{U} \) are in \( \mathbb{F}_{q^{(b+c)/d}} \) and since \( w_{i,j} \) are in \( \mathbb{F}_q \) for all \( i \) and \( j \), it follows that \( \sum_{j=1}^{(b+c)/d} w_{i,j}u_j \) is an element of \( \mathbb{F}_{q^{(b+c)/d}} \) for all \( i \). Therefore, since \( \mathcal{V} \) is a basis of \( \mathbb{F}_{q^{b+c}} \) over \( \mathbb{F}_{q^{(b+c)/d}} \), (9.17) implies that

\[
\sum_{j=1}^{b+c/d} w_{i,j}u_j = 0 \tag{9.18}
\]

for all \( i \in [d] \). Further, since \( \mathcal{U} \) is a basis of \( \mathbb{F}_{q^{(b+c)/d}} \) over \( \mathbb{F}_q \), and since \( w_{i,j} \in \mathbb{F}_q \), (9.18) implies that \( w_{i,j} = 0 \) for all \( i \in [d] \) and for all \( j \in \lfloor \frac{b+c}{d} \rfloor \). \( \blacksquare \)
Discussion
The contribution of this dissertation is twofold. On one hand, mathematical aspects of subspace codes were studied, giving emphasis to structural notions of interest. On the other hand, several problems regarding distributed storage codes were studied, giving emphasis to their connection to subspace codes. In what follows, we briefly summarize our contributions while pointing out future research directions.

Regarding equidistant subspace codes, our main contribution is a construction of such codes using the Plücker embedding and Steiner systems. A Steiner system $S(t, k, n)$ is a pair $(Q, B)$, where $Q$ is an $n$-set and $B$ is a collection of $k$-subsets of $Q$ called blocks, such that any $t$-subset of $Q$ is contained in precisely one block. The Plücker embedding is a unique way to map subspaces in the Grassmannian $\mathcal{G}_q(n, k)$, the set of all $k$-subspaces of $F_q^n$, into points in the projective plane $\mathbb{P}^{(n-1)}_q$. In a nutshell, in our work we consider the Steiner system $S(2, q + 1, \frac{q^n - 1}{q - 1})$ that is obtained for any prime power $q$ from having $\mathcal{G}_q(n, 1)$ as the set $Q$ and $\mathcal{G}_q(n, 2)$ as the blocks $B$. By applying the Plücker embedding of this set of blocks, we obtain a nontrivial equidistant subspace code in $\mathcal{G}_q\left(\binom{n}{2} n - 1\right)$ of size $\frac{q^n - 1}{q - 1}$. This construction is applicable to any given field $F_q$ and is one of the largest constructions to date.

Other than providing this construction, bounds and trivial construction were discussed. The notions of a sunflower and a ball were introduced. A sunflower is an equidistant subspace code in which any two codewords meet in the same subspace. A ball is an equidistant subspace code in which all ($k$-dimensional) codewords are contained inside a subspace of dimension $k + 1$. In our work, these two notions are defined as trivial, sunflower codes are shown to follow from the notion of (partial) spreads, and balls are shown to optimal for their respective parameters.

As for bounding the maximal possible size of an equidistant subspace code, a former conjecture by Deza was proven false by a computer search, and a general upper bound was derived from a parallel one in the Hamming space. Our Plücker construction attains the conjectured bound by Deza, and thus it is not optimal, at least for the parameters for which the counterexample was found.

It is worth noting that since our paper was published, a few advances were obtained by other groups of researchers. In particular, [2] employed advanced projective geometric techniques and obtained a larger counterex-
ample for the same parameters. Moreover, [13] proved that an equidistant code of maximum cardinality is always either a sunflower of a complement of a sunflower, given that the underlying field is large enough, resolving a problem which was left unanswered in our paper. Yet, closing the gap between the upper bound for nontrivial equidistant subspace codes which is given in our paper, and the known constructions, is still an open problem.

Next, the notion of cyclic subspace codes was studied. This notion was introduced by several previous authors [10, 33], both as a means to construct subspace codes over any field and as a study of automorphism groups of subspace codes. A subspace code \( C \) in \( G_q(n, k) \) is called cyclic if it is closed under multiplication by a nonzero element of \( \mathbb{F}_{q^n} \). That is, for any \( V \in C \) and any \( \alpha \in \mathbb{F}_{q^n}^* \), we have that \( \alpha V \in C \), where \( \alpha V \triangleq \{ \alpha v | v \in V \} \). The simplest example of such code, and the only general one which was known until our work, is the set of all cyclic shifts of any subfield. Since this example is of minimum distance \( 2k \) and size \( \frac{q^n - 1}{q^k - 1} \), it was conjectured in [33] that for any \( q, k \), and a large enough \( n \), there exist a cyclic subspace code in \( G_q(n, k) \) with minimum subspace distance \( 2k - 2 \). This conjecture was verified in our paper, along several possible explicit constructions.

In our work we observed that given a subspace \( V \in G_q(n, k) \), the properties of the orbit of \( V \), i.e., the set \( \{ \alpha V | \alpha \in \mathbb{F}_{q^n} \} \), follow in some cases from the subspace polynomial of \( V \). A subspace polynomial of a given subspace is a unique monic polynomial of lowest degree whose set of roots is \( V \), and this polynomial is known to be a linearized polynomial. That is, a large gap between the non-zero coefficients induced a large distance between \( V \) and its cyclic shifts, and non-zero coefficients in certain positions induce a large orbit. Since any monic linearized polynomial is a power of a subspace polynomial in its splitting field, these properties allowed us to resolve the aforementioned conjecture by considering certain linearized polynomials, and defining the subspace codes as the orbit of their set of roots. Further, we managed to add more orbits to that code by the well-known Frobenius automorphism, a technique that was later greatly improved by [21].

The main drawback in our work, whose improvement is one of our current research goals, is the fact that the degree of the splitting field is in general not known. As a result, the relation between \( n \) and \( k \) is not known in this construction, and is usually unreasonably large. To this date, it seems that
no progress in this direction was achieved.

Studying cyclic subspace codes with \textit{degenerate} (that is, smaller than the optimal) orbits has led us to several conclusions about the coefficient structure of subspace polynomials. This seemingly esoteric discovery was soon to be discovered to have applications in list decoding of Gabidulin codes. In several past past works \cite{35,4}, subspace polynomials were used to show the limits of list-decoding Reed-Solomon and Gabidulin codes. Consequently, several authors have raised the question of list decodability of Gabidulin codes themselves, as oppose to their subcodes or variants. Our discovery has resolved this question in the negative sense, that is, there exists infinite families of Gabidulin codes which cannot be list decoded efficiently at all beyond the unique decoding radius.

A drawback in this work, that was unfortunately discovered only after the paper was published, is that its central theorem applies only to square Gabidulin codes, and Gabidulin codes in $\mathbb{F}_{q^m}$ such that $n \mid m$ and the evaluation points reside in a cyclic shift of $\mathbb{F}_{q^n}$. Removing this restriction seems to correlate strongly with the following open question. Given a subspace $V$ and a nonsingular linear transform $A$, what can we say about the coefficients of the subspace polynomial of $AV \triangleq \{ Av | v \in V \}$, in comparison with the coefficients of the subspace polynomial of $V$? For the case in which $A$ is either a multiplication by a field element\footnote{That is, $A$ is a power of the companion matrix of the modulus polynomial of $\mathbb{F}_{q^n}$.} or a Frobenius automorphism, an answer follows from our paper \cite{3}. For a general $A$ which has neither of those forms, this question seems to be wide open.

Our next contribution was the construction of Minimum Storage Regenerating (MSR) codes over small fields from $(A,S)$-sets. This work includes the first explicit construction as access-optimal and non access-optimal MSR codes for more than two parity nodes, an MSR codes for two parities in which the previously obtained field size is halved. Besides being explicit, the codes for three parities also obtain an exponential reduction in field size in comparison with the existential constructions which preceded it.

Construction of high-rate MSR codes attracted a considerable amount of attention in the past year, with new groundbreaking results appearing on a near-monthly basis. In particular, the once widely open problem of obtaining explicit high rate MSR codes seems to be resolved almost completely in \cite{12,29,36,37}. Moreover, the authors of this work chose not to use the
convenient notion of an \((A,S)\)-set to construct the generator matrix, but use variants of it which are induced by considering the parity check matrix instead.

Our study of sets of permutations with locality was the first of its kind, and thus naturally, has no predecessors for comparison. In this work, we established the connection between Locally Recoverable Codes and permutation codes, proved several upper bounds and provided trivial optimal construction for a small set of parameters. For parameter sets in which the upper bounds are not achieved, constructions are provided using various mathematical tools such as Reed-Solomon codes and permutation polynomials.

Existential lower bound follow in one of two ways. One way is a simple counting argument, which states that an optimal set of permutations with locality exists in some coset of an optimal locally recoverable code over the alphabet of integers modulo \(n\). The other, more intriguing way follows from an algorithmic understanding, which is yet to be known, of a variant of the non-attacking queens problem. In this variant, queens are replaced by semi-queens, which cannot move in the north-east south-west diagonal, and the ordinary chessboard is replaced by a toroidal one, in which movement is allowed on generalized diagonals. The number of distinct non-attacking queens configurations was only very recently estimated, and is in general not known. It is shown in our work that an efficient algorithm for obtaining the \(i\)-th configuration may be used for construction of an optimal set of permutations with locality, and in some sense, also vice versa. Given this equivalence, it is not unreasonable to assume that an optimal construction of such set of permutations with locality is unlikely to be found using current techniques.

Moreover, as some reviewers of this work noted, this problem is much more of a mathematical interest than of a practical one. While storage of a permutation might greatly assist the updating process of systems which store constantly-permuting data, storing a permutation using an ordinary locally recoverable or a regenerating code seems to be adequate in most conceivable scenarios. In particular, sets of permutations with locality do not necessarily include any particular permutations of interest, and their construction seems to be a highly involved process which attains a negligible improvement over the trivial solutions.
Finally, our work discussed the application of subspace codes to distributed storage. In [23], it is shown that the equidistant codes which we constructed in [9] using the Plücker embedding may be used to obtain distributed storage systems near the minimum bandwidth point of the cut-set tradeoff. In this construction, each storage node is associated with a codeword (subspace) in the subspace code, and stores a projection of the file on that subspace, that is, a multiplication of the file by some spanning matrix of the subspace.

As previously noted, this construction does not attain better performance than the well known product matrix codes [22], and are also applicable to only a small set of possible parameters. Yet, this work presents a fascinating connection between algebraic geometry and distributed storage. Further, when reopening this topic nearly two years later, it was discovered that combining subspace codes with product matrix codes enables to achieve asymptotically optimal regenerating codes over any field, in the minimum bandwidth as well as the minimum storage point. That is, each node is still associated with a subspace from a certain subspace codes, and stores not a projection of the file on that subspace, but a projection of some symmetric matrix on it. That is, the file is transformed into a symmetric matrix, as in [22], and only then the projection is computed. This symmetrizing operator enables repair and reconstruction algorithms which highly resemble those of [22], where non-commutativity issues between submatrices are resolved using the properties of the Kronecker product.

Studying this topic further, a more convenient technique to present these asymptotically optimal codes was discovered. This technique involves representation of extension fields as sets of matrices, as well as cyclotomic cosets. Nevertheless, the original construction using subspace codes is given in this dissertation for the sake of the complete representation of the progress of the research.

At this point, it is yet unclear what is the complexity of decoding the suggested codes in comparison with the original product matrix codes. While both codes seem to have a similar complexity when decoding naïvely using matrix multiplication, both codes have an underlying Vandermonde structure which may aid in future research of this topic.
Bibliography


