OPTIMAL ADDITIVE QUATERNARY CODES OF DIMENSION 3.5

ABSTRACT. After the optimal parameters of additive quaternary codes of dimension $k \le 3$ have been determined in [2], there is some recent activity to settle the next case of dimension $k = 3.5$ [8, 9]. Here we complete dimension $k = 3.5$ and give partial results for dimension $k = 4$.

Keywords: additive codes, linear codes, quaternary codes, Galois geometry

Mathematics Subject Classification: 94Bxx, 51E22

1. INTRODUCTION

A quaternary block code C of length n is a subset of $\mathbb{F}_4^n$. If C is closed under componentwise addition then C is called additive. If C is additive and closed under $\mathbb{F}_4$ scalar multiplication then C is called linear. The parameter k such that the number of codewords $|C|$ equals 4^k is called the dimension of C (in both special cases). Clearly, k is an integer if C is linear and a half-integer if C is additive. For each integer s let $n_k(s)$ denote the maximal length n such that an additive quaternary code of length n, dimension k, and minimum Hamming distance $n - s$ exists. For $k \le 3$ the function $n_k(s)$ was completely determined in [2]. In the sequence of papers [8, 9] the determination of $n_{3.5}(s)$ was narrowed down to $s \in \{6, 7, 12\}$.[1] Geometrically, $n_k(s)$ is the maximum number of lines in the projective space $\mathrm{PG}(2k-1, 2)$ such that each hyperplane contains at most s lines, which corresponds to a binary linear code of length $3n_k(s)$, dimension $2k$, and minimum Hamming distance $2(n_k(s) - s)$ in coding theory terms[2] if we replace each line by its contained three points, see [2]. For $k \le 3.5$ and $s \ge 4$ the known optimal parameters of binary linear codes imply the correct upper bounds for $n_k(s)$. The small cases of s that are covered in [4]. Taking the union of two multisets of lines implies $n_k(s_1 + s_2) \ge n_k(s_1) + n_k(s_2)$ and $n_k(s) \ge n_k(s-1) + 1$. So, for $k = 3.5$ we only need constructions for $s \in \{3, \dots, 13, 15, 21, 25, 26, 30, 31\}$ as base examples. Except for $s \in \{6, 13\}$ examples can easily be found by prescribing a group of order 3 or 5 as a subgroup of the automorphism group and integer linear programming. For the two other cases we have used `LinCode`[5] to exhaustively generate linear binary codes as candidates whose corresponding multisets of points are then partitioned into lines.[3] As a compact representation we sort the columns of a generator matrix such that each consecutive triplet of columns corresponds to the three points of a line. Replacing three consecutive bits a_1, a_2, a_3 by $4a_1 + 2a_2 + a_3$ yields the following matrices

$$\begin{pmatrix} 0000003333333333333333 \\ 0333330000055555533333 \\ 0055550555305536600533 \\ 3005333563300003655655 \\ 0556363003530036503556 \\ 5636550363330330060650 \\ 3300355365056036053530 \end{pmatrix}, \quad \begin{pmatrix} 0000003333333333333333330 \\ 0003330005555555555553330 \\ 3335555550003333336660000 \\ 5355360335365353660330553 \\ 3563653303653636553305505 \\ 5066350555633360600556600 \\ 0663565506353006635506060 \end{pmatrix},$$

$$\begin{pmatrix} 00000000033333333333333333333333330003333333 \\ 00033333300055555555555555555553333330330555555 \\ 33355555550005555553333333666666555555353505553 36 \\ 000555666555000333000555333666555666505305603 5 \\ 55555533300033333366600066600066633306363053 60 \\ 53653656356353656356353656353656356300000000000 \\ 36536563563536563563536563536563563500000000000 \end{pmatrix}, \quad \text{and}$$

$$\begin{pmatrix} 000303333333330333333300000033333333333333333333333 \\ 030303300330330030300333333555555555555555555555555 \\ 303330003033335555555555555503033000033655655656566 5 \\ 333035555555553000300356665600303056556003055566556 6 \\ 030550330656630335555356666300055333030566033005666 \\ 335053356035633660355300055006505306556066300663335 \\ 550636533663536330535600633055600065533550050633600 \end{pmatrix}$$

for $s = 6$, 7, 12, and 13, respectively.

---

[1] The example for $s = 13$ refers to [10].

[2] Additionally, all occurring weights are even and the maximum weight is $2n_k(s)$.

[3] There are a unique $[66, 7, \{32, 34, \dots, 44\}]_2$- and two $[153, 7, \{76, \dots, 102\}]_2$-codes.

In order to complement [9] we give geometric constructions for the other base cases in Section 2. We give partial results for dimension $k = 4$ in Section 3.

2. Geometric constructions

Points in $\mathrm{PG}(k-1, 4)$ map to lines in $\mathrm{PG}(2k-1, 2)$. Taking a subcode of dimension one less geometrically corresponds to the projection through a point P. Each line containing P is mapped to a double-point Q and may replaced by an arbitrary line containing Q. Starting from a $\mathbb{F}_4$-linear code and the corresponding multiset of points, base examples for e.g. $k = 3.5$, $s \in \{5, 21\}$ can be obtained this way.

A vector space partition of type $1^{t_1}2^{t_2}\dots$ is a collection of subspaces that partition the set of points such that exactly t_i of these subspaces have dimension i, see e.g. [6]. It is well known that for each pair of integers, satisfying $0 \le a < b$ and $a \equiv b \pmod 2$, there exists a vector space partition of $\mathrm{PG}(b-1, q)$ consisting of $t_2 = q^a \cdot \frac{q^{b-a}-1}{q^2-1}$ lines and a single a-dimensional subspace A. Each hyperplane H contains $q^{a-2} \cdot \frac{q^{b-a}-1}{q^2-1}$ lines if $A \not\le H$ and q^{a-2} less otherwise. If $a = 0$ then we also speak of a line spread. Vector space partitions of types $2^{40}3^1$, $2^{35}3^14^1$ [6], and $2^{32}5^1$ give base examples for $k = 3.5$, $s \in \{8, 9, 10\}$ by removing the subspaces that are not two-dimensional.

Let $\mathcal{L}_1$ be a multiset of n_1 lines in $\mathrm{PG}(k-1, q)$ and A be an a-dimensional subspace such that each hyperplane H contains at most s_0 lines if $H \ge A$ and at most s_1 lines otherwise. Let $\mathcal{L}_2$ be a multiset of n_2 lines in $\mathrm{PG}(a-1, q)$ such that each hyperplane contains at most s_2 lines. Then, taking the multiset union of $\mathcal{L}_1$ and $\mathcal{L}_2$ with a suitable embedding of $\mathrm{PG}(a-1, q)$ as A gives a multiset of $n_1 + n_2$ lines in $\mathrm{PG}(k-1, q)$ such that each hyperplane contains at most $\max\{s_1 + s_2, s_0 + n_2\}$ lines. Applying this construction to $\mathcal{L}_1$ arising from a vector space partition of type $2^{40}3^1$ and three lines different lines in $\mathrm{PG}(2, 2)$ as $\mathcal{L}_2$ gives a base example for $(k, s) = (3.5, 11)$.

The existence of a vector space partition of type $2^{32}5^1$ such that 4 lines are contained in a 4-dimensional space A is not hard to show. Taking the union with a second such example that contains the line missing in A and removing the five lines from A gives a base example for $(k, s) = (3.5, 15)$.

Let l be a positive integer, H be a hyperplane of $\mathrm{PG}(l+2, q)$, and A be a l-dimensional subspace of H. By $B_1, \dots, B_{q+1}$ we denote the $(l+1)$-dimensional subspaces with $A \le K_i \le H$. Partition the set of all points except those from K_i by lines and denote the multiset union of lines of these $q+1$ vector space partitions of type $2^{t_2}(l+1)^1$ by $\mathcal{L}^\star$. If l is even we denote by $\mathcal{L}_A$ a line spread of A and by $\mathcal{L}_H$ a line spread of H. The multiset union of $\mathcal{L}^\star$, $\mathcal{L}_A$, and q copies of $\mathcal{L}_H$ consists of $\frac{q^{l+3}-1}{q-1}$ lines and covers each point exactly $q+1$ times.[4] The construction allows to remove $\mathcal{L}_A$, copies of $\mathcal{L}_H$, or subsets thereof in any combination.[5] Choosing $l = 4$ for $k = 3.5$ gives base examples for $s \in \{25, 26, 30, 31\}$ as well as examples for $s \in \{19, 20, 21, 24, 28, 29\}$.

For $k = 3.5$ and $s \le 4$ we refer to [4].

3. Partial results for dimension 4

In Table 1 we state the known bounds for $n_4(s)$. Lower bounds based on quaternary linear codes are stated in columns headed with "L". Upper bounds, based on [4] for $s \le 4$ and on binary linear codes for $s > 4$, are stated in columns headed with "U". Values of improved constructions are given in columns headed with "I". Open cases are marked in bold font and we remark that we have $n_4(s) = n_4(s-21) + 85$ for $n > 60$. For $s > 60$ there are improvements over the linear case iff s is congruent to 2, 3, 7, or 8 modulo 21. Generator matrices of the improvements are given in Section A. We observe that $n_4(44) \ge n_4(23) + n_4(21)$ is attained with equality and that there are easy geometric constructions for $s \in \{49, 50\}$.

Choosing $l = 5$ for $k = 4$ yields a multiset $\mathcal{L}^\star$ of 160 lines with $s = 40$. Consider hyperplane H of the construction as $\mathrm{PG}(6, 2)$ and insert the lines from a vector space partition of type $2^{32}5^1$. This yields a multiset of 192 lines with $s = 48$. Now consider the special subspace A of the construction as $\mathrm{PG}(4, 2)$ and insert either three lines in a three-dimensional subspace or the lines from a vector space partition of type 2^83^1. This yields examples for $s \in \{49, 50\}$.[6]

References

[1] S. Ball, M. Lavrauw, and T. Popatia. On additive codes over finite fields. *arXiv preprint 2406.08916*, 2024.

[2] J. Bierbrauer, S. Marcugini, and F. Pambianco. Optimal additive quaternary codes of low dimension. *IEEE Transactions on Information Theory*, 67(8):5116–5118, 2021.

[3] J. Bierbrauer, S. Marcugini, and F. Pambianco. An asymptotic property of quaternary additive codes. *Designs, Codes and Cryptography*, 92:3371–3375, 2024.

[4] A. Blokhuis and A. E. Brouwer. Small additive quaternary codes. *European Journal of Combinatorics*, 25(2):161–167, 2004.

[5] I. Bouyukliev, S. Bouyuklieva, and S. Kurz. Computer classification of linear codes. *IEEE Transactions on Information Theory*, 67(12):7807–7814, 2021.

[4] In [3] a 3-cover was used to construct asymptotically optimal quaternary additive codes.

[5] Note the similarity to the removal of subspaces in the construction of Solomon and Stiffler for codes meeting the Griesmer bound [7, 11].

[6] Actually, the constructions from Section 2 are sufficient to attain the maximal number $n_k^q(s)$ of lines in $\mathrm{PG}(2k-1, q)$ such that at most s lines are contained in a hyperplane, assuming that s is sufficiently large. I.e., the upper bound implied by the Griesmer bound can always be attained if s is sufficiently large, c.f. [3]. We do not know what happens if we replace lines by subspaces with a larger dimension, see e.g. [1].

s	L	I	U	s	L	I	U	s	L	I	U
1	–		–	21	85		85	41	165		
2	–		–	22	86		86	42	170		
3	5		5	23	87	89	89	43	171		
4	10		10	**24**	**92**		**94**	44	172	174	174
5	17		17	25	97		97	45	177	179	179
6	18		18	26	102		102	46	182		
7	23		23	**27**	**103**	**106**	**107**	47	187		
8	28		28	**28**	**108**		**110**	48	192		
9	31	33	33	**29**	**113**		**115**	49	193	195	195
10	**34**	**35**	**36**	30	118		118	50	198	200	200
11	39	40	40	31	123		123	51	203		
12	44		44	32	128		128	52	208		
13	49		49	33	129		129	53	213		
14	50	54	54	34	134		134	54	214		
15	**55**		**57**	35	139		139	55	219		
16	64		64	36	144		144	56	224		
17	65		65	37	149		149	57	229		
18	70		70	38	150		150	58	234		
19	75		75	39	155		155	59	235		
20	80		80	40	160		160	60	240		

TABLE 1. Bounds for $n_4(s)$.

[6] S. El-Zanati, G. Seelinger, P. Sissokho, L. Spence, and C. Vanden Eynden. On partitions of finite vector spaces of low dimension over $\mathrm{GF}(2)$. *Discrete Mathematics*, 309:4727–4735, 2009.

[7] J. H. Griesmer. A bound for error-correcting codes. *IBM Journal of Research and Development*, 4(5):532–542, 1960.

[8] C. Guan, R. Li, Y. Liu, and Z. Ma. Some quaternary additive codes outperform linear counterparts. *IEEE Transactions on Information Theory*, 2023.

[9] C. Guan, J. Lv, G. Luo, and Z. Ma. Combinatorial constructions of optimal quaternary additive codes. *IEEE Transactions on Information Theory*, pages 1–12, to appear.

[10] S. Kurz. Computer classification of linear codes based on lattice point enumeration. In *International Congress on Mathematical Software*, pages 97–105. Springer, 2024.

[11] G. Solomon and J. J. Stiffler. Algebraically punctured cyclic codes. *Information and Control*, 8(2):170–179, 1965.

Appendix A. Generator matrices for dimension 4

Here we state the found examples improving $n_4(s)$ over $\mathbb{F}_4$-linear codes. Starting from the compact representation of generator matrices introduced in Section 1 we apply the transformations $0 \to 00$, $3 \to 01$, $5 \to 10$, $6 \to 11$ and convert blocks of four bits to hexadecimal notation afterwards.

$s = 10$:

```
111044411555555554
115555555208AAAAA8
552AA29641A643AAFC
A4921BCA4923CD27E0
E438571EA612608590
413253688DCEECD968
D8F3215BCC62D5864C
C25ED8DF9C07E3360C
```

$s = 11$:

```
10411110455555555555
10555555542002AAAAAA
1542AAA5901A65402BBF
51098FF290919AA64FF0
AA1922185214F7AF80B7
1DAC26DA5553DC0A3C13
654BE1CE2720412FFD3B
F878CACE099AD75B0170
```

$s = 14$:

```
000555555555555555555555555
555111111AAAAAAAAAAAAAAAAAA
99A19A465333311112222AAFF55
66F56D1B4BBBB5555EEEE000000
000928AA6B7FD328C9A6E271BAF
0006E3D8B8A97BE326DFCBDE623
8536B12E514AE590E48D3F84F2F
347A6503EEA69F80C4D95A3D0C7
```

$s = 23$:

```
000000000015555555555555555555555555555555554
00005555555554000000AAAAAAAAAAAAAAAAAAA555554
555500000AA400025502AAAAAA9555FFFFFFFFFAAAAA8
AAAAAAAA9AF800A7A914257FFFC03F002A57FFFAA5554
00AA00ABEF5AAAB89E4E9540A5EBCF0BCF00A55A9FFFC
25051F28B10E5FA4F92092E9906CFF1ADB0A3AB2E297C
817B324A142F2710ABB3156FA277E79CEC98C28CFCE64
6CC371E3A2409352BF53A348DB8BF6BC324AC49BE7AD0
```

$s = 27$:

```
00000000000005555555555555555555555555555555555555551
000000055555511111111111AAAAAAAAAAAAAAAAAAAAAAAAAAAA2
01111559999AA9999AAAA6666000333311112222AAFF99A55F5F0
05555006666FF6666DDDDBBBB000BBBB5555EEEE000066F00AF50
59999AA99660599EE55CC88669B7889988881100AF05FFFAAF000
A2222AA33AA0A3399AACC88116ED88DD6666BBEE00FFCC5FFA550
30B5EAF9F359906062F85385E5AF5E7C287D2CB5A50F249058944
E0D49B1690FEB0FB452DA5849183F74CB6F2596A72960F1B1B888
```

$s = 44$:

```
000000000000000000005555555555555555555555555555555555555555555555555555555555555055
0000000555555555555555000000000000000AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA555555555555550A
55555550002AAA55555550002AAAAAA955500000000002AAAAAA95555555555FFFFFFF0002AAAAAA9555A22
AAAAAAA5557FFFAAAAAAAA55555555556AAA0002AAA5556AAAAAA8000AAAAAAA0003FFFAAABFFFFFFC0007BB
AAAAAAA5557FFF0001555AAAAAAAAAA8000FFFC0005554000FFFEAAA5555555AAAAAAAAAA80000002AAABF0
12E997F29686E73B949B71A7C4EBA57CEE52E5C6E71B9C69F1A7C69F13AE95F1A7CEE51A7C49A6DFC69F000
9E57F28629F47AB5C9BD897E26D5FCA2D72E5CB47AD1EA5F897E25F89B57F2897E2D7297E25B7F6825F8000
6B857CE94EDF29ED266F161F9B8D1E7BB49B971F297CA587E61F987E6E3479E61FBB4961F9A61FCE587E000
```

$s = 45$:

```
000000000000005555555555555555555555555555555555555555555555555555555555555555555155554
555555555555550000000000000000000000AAAAAAAAAAAAAAAAAAAAAAAAAAAA555555555555555555555540AA54
0000000AAA95550000000000AAAAAAAAAA95550000000AAAAAAAFFFFFFFFFFFFFAAAAAAAAAAAAAAAAA95558A2928
AAA95550002AAA00015555555555FFFEAAA00015550002AAA0000000FFFFFFFAAA9555FFFFFFFFFFEAAA0B5F88
5556AAAAAA8000AAA80005555555FFFEAAAFFFC000AAAAAAA55555550002AAA0002AAA0001555FFFFFFF2387E4
1B9C69F399CEE51B9C6A913AE95F1A7C69F2E5CE671A7C69F13AE95F26DCEE51A7C69F26DC69F1A7C69F000000
D1EA5F8736AD72D1EA47E9B57F2897E25F8E5C9CDA97E25F89B57F286F62D7297E25F86F625F897E25F8000000
7CA587ED4BBB497CA5B266E3479E61F987EB97352E61F987E6E3479E9BC7B4961F987E9BC587E61F987E000000
```

$s = 49$:

```
00000000000000000000555555555555555555555555555555555555555555555555555555555555555555540055555554
0000000000155555555550000000000AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA5555555555555500AAA554
00015555556AAAAAAAAAAAAAAAAA0000000000000AAAAAAAAAAAAAA5557FFFFFFFFFFFFFFFFFAAAAAAAAAAAAA9290AF2A8
5554000AAA80005557FFFAAA955500015555557FFF0000000AAA9555FFFC000AAAAAAAAAABFFFAAAAAAAAFFFC2986B2787C
AAAAAAAAAA8000AAA9555AAAAAAA5556AAAFFFFFFFAAA955555555550003FFF0002AAA5556AAA0003FFF0001CAB374F58C
1A7CB972E5CE671A7C6E73B9469F1A7CE67399C69F1A7C69F26DC69F1A7CEE51A7C69F1A7CEE51A7C69F26DC0000000000
97E3972E5C9CDA97E347AB5CA5F897E1CDA736A5F897E25F86F625F897E2D7297E25F897E2D7297E25F86F600000000000
61FAE5CB97352E61F9F29ED2587E61FB52ED4B987E61F987E9BC587E61FBB4961F987E61FBB4961F987E9BC40000000000
```

$s = 50$:

```
000000000000000000005555555555555555555555555555555555555555555555555555555555555555555554001555555555
0000000555555555555555000000000000000AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA95555555555555415402AAAAA55
55555550002AAAAAA95550000000AAA955500000000002AAA55555555557FFFFFFFFFFFFEAAAAAAAAAA55546A4A40A55F2A
AAAAAAA0002AAA5554000AAA95555556AAA0000000AAA9555AAABFFFFFFC000AAAAAAA5554000AAABFFFAAA94BA5AAF09109
AAA9555AAABFFFAAAAAAAFFFEAAAFFFD5550003FFFAAAAAAAAAA8000FFFEAAAAAABFFF00015550000000FFFE976BA7224C8E
2E5C9B726DC9B71A7C69F1A7C69F1A7C69F2E5CB9726DC69F1A7C69F1A7C69F1A7C69F1A7C9B71A7C9B71A7C000000000000
E5C9BD86F61BD897E25F897E25F897E25F8E5CB9726F625F897E25F897E25F897E25F897E1BD897E1BD897E000000000000
B9726F19BC66F161F987E61F987E61F987EB972E5C9BC587E61F987E61F987E61F987E61FA6F161FA6F161F800000000000
```