Designs, Codes and Cryptography manuscript No.
(will be inserted by the editor)

Classification and nonexistence results for linear codes with prescribed minimum distances

Thomas Feulner

Received: date / Accepted: date

Abstract Starting from a linear $[n, k, d]_q$ code with dual distance $d^\perp$, we may construct an $[n-d^\perp, k-d^\perp+1, \geq d]_q$ code with dual distance at least $\left\lceil \frac{d^\perp}{q} \right\rceil$ using construction Y_1. The inverse construction gives a rule for the classification of all $[n, k, d]_q$ codes with dual distance $d^\perp$ by adding $d^\perp$ further columns to the parity check matrices of the smaller codes. Isomorph rejection is applied to guarantee a small search space for this iterative approach.

Performing a complete search based on this observation, we are able to prove the nonexistence of linear codes for 16 open parameter sets $[n, k, d]_q$, $q = 2, 3, 4, 5, 7, 8$. These results imply 217 new upper bounds in the known tables for the minimum distance of linear codes and establish the exact value in 109 cases.

Keywords Classification · code equivalence · construction Y_1 · linear code · residual code · semilinear isometry

Mathematics Subject Classification (2000) 94B65 · 94B05 · 05E18

1 Introduction

Let $\mathbb{F}_q^n$ denote the n-dimensional vector space over the finite field $\mathbb{F}_q$. The *Hamming weight* $\mathrm{wt}(v)$ of a vector $v \in \mathbb{F}_q^n$ is the number of its nonzero entries. The Hamming distance $\mathrm{d_H}(u, v) = \mathrm{wt}(u - v)$ counts the number of positions where the vectors $u, v \in \mathbb{F}_q^n$ are different. A linear $[n, k, d]_q$ code is a k-dimensional subspace of $\mathbb{F}_q^n$ such that any two different codewords $c, c' \in C$ have *Hamming distance* $\mathrm{d_H}(c, c')$ at least d and there is a pair whose distance is equal to d. The parameter d is called the *minimum distance* of the code C and is equal to the minimum weight of the nonzero codewords.

The research of the author is supported by Deutsche Forschungsgemeinschaft under contract WA 1666/7-1 within the Schwerpunktprogramm 1489.

T. Feulner
Department of Mathematics, University of Bayreuth, 95440 Bayreuth, Germany
E-mail: thomas.feulner@uni-bayreuth.de

Any matrix $\Gamma \in \mathbb{F}_q^{k\times n}$ whose rows form a basis of the $[n,k,d]_q$ code C is called a *generator matrix* of C. Conversely, any matrix $\Delta \in \mathbb{F}_q^{(n-k)\times n}$ of full row rank with

$$C = \{w \in \mathbb{F}_q^n \mid w \cdot \Delta^{\mathrm{T}} = 0\} \tag{1}$$

is called a *parity check matrix* of C. Equation (1) shows that an $(n-k)\times n$-matrix Δ of full row rank defines an $[n,k,d]_q$ code if and only if there exists d columns that are linearly dependent and any $d-1$ columns of Δ are linearly independent. The *dual code* $C^\perp$ of a linear code C is the subspace of vectors that are orthogonal to C under the standard bilinear form, i.e.

$$C^\perp := \{w \in \mathbb{F}_q^n \mid \forall c \in C : \langle c, w\rangle = 0\}.$$

The parity check (generator) matrices of a code C are exactly the generator (parity check) matrices of $C^\perp$. We say that C is an $[n,k,d]_q^{d^\perp}$ code if C is an $[n,k,d]_q$ code and its dual code $C^\perp$ has minimum distance $d^\perp$.

A mapping $\iota : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is called an *isometry*, if it respects the Hamming metric. Two codes C, C' which are related to each other via the application of an isometry ι, i.e. $C' = \iota(C)$, have the same error-correcting capability.

In general, subspaces are not preserved under some arbitrary isometry. Hence, we engage in the most general subgroup under this additional condition, which is for $n \geq 3$ equal to the group of semilinear isometries [2]. A mapping $\iota : \mathbb{F}_q^n \to \mathbb{F}_q^n$ is called *semilinear*, if there exists an automorphism α of $\mathbb{F}_q$ such that, for all $u, v \in \mathbb{F}_q^n, \kappa \in \mathbb{F}_q$ the following holds $\iota(u+v) = \iota(u) + \iota(v)$ and $\iota(\kappa u) = \alpha(\kappa)\iota(u)$.

The isometry ι of $\mathbb{F}_q^n$ is representable by some monomial matrix $M \in \mathrm{GL}_n(\mathbb{F}_q)$ and a field automorphism α, i.e.

$$\iota(v) = \alpha(v)M^{\mathrm{T}}, \text{ for all } v \in \mathbb{F}_q^n,$$

where α is element-wisely applied to the vector v. We also write $\alpha(\Delta)$ for the element-wise application of α to a matrix Δ over $\mathbb{F}_q$.

Two codes C, C' are *equivalent*, if there exists a semilinear isometry $\iota : \mathbb{F}_q^n \to \mathbb{F}_q^n$ with $\iota(C) = C'$. Two parity check matrices $\Delta, \Delta' \in \mathbb{F}_q^{(n-k)\times n}$ are called equivalent, if they form parity check matrices of equivalent codes.

Proposition 1 *C is equivalent to C' if and only if $C^\perp$ is equivalent to $C'^\perp$. Hence, if C is an $[n,k,d]_q^{d^\perp}$ code, so is C'.*

Proof The semilinear isometry ι with $\iota(C) = C'$ is representable by some monomial matrix $M \in \mathrm{GL}_n(\mathbb{F}_q)$ and a field automorphism α. It is easy to prove that $\iota^\perp(v) := \alpha(v)M^{-1}$ also defines a semilinear isometry on $\mathbb{F}_q^n$ and that $\iota^\perp(C^\perp) = C'^\perp$. □

The goal of the paper is to classify all linear codes with given parameters $[n,k,\geq d]_q^{d^\perp}$ up to semilinear isometry. This means that we compute a set T of parity check matrices representing all nonequivalent codes with these parameters. Within this work, we just attack the problem of calculating reasonable small sets $\mathcal{T}$ which contains the set T, see Section 3.1. Afterward, the algorithm [6] is used to compute a *unique* representative of the equivalence class for each element of the candidate set $\mathcal{T}$. Hence, we are able to remove equivalent codes and we are left with T, for more details see Section 3.2.

The paper is organized as follows. The next section repeats some elementary constructions to derive new codes from old and gives their parameters. The main algorithm is developed in Section 3. It is based on construction Y_1, which is inverted. This method adopts the approaches described in [4], [5] to be used for a full classification of linear codes over arbitrary finite fields. Our contribution compared to [5] is the fact that we provide a full classification. The algorithm described in [4] is limited to binary codes. We have generalized this approach allowing arbitrary fields and semilinear isometry. Furthermore, we combined the ideas of their procedures BRUTEFORCE and EXTEND in one procedure.

The rest of the paper states the results and consequences on the tightening of upper bounds on the minimum distances. Furthermore, we give an application [7] of the algorithm for proving the nonexistence of an automorphism of order 7 for a self-dual binary $[72, 36, 16]$ code.

Within this work, we suppose some total ordering on the finite field $\mathbb{F}_q$, where $0 < 1 < \mu$, for all $\mu \in \mathbb{F}_q \setminus \{0, 1\}$. Vectors are ordered colexicographically. Furthermore, matrices are interpreted as lexicographically ordered tuples of vectors. The notation $\boldsymbol{\mu_n}$ with $\mu \in \mathbb{F}_q$ denotes the all-μ vector of length n.

2 Code constructions

In this section we describe several methods to derive new codes from old. We will also give their parameters in terms of the old.

Definition 1 (Puncturing) The *puncturing* of a linear code C in a coordinate $i \in \{0, \ldots, n-1\}$ is a linear code of length $n-1$ achieved by eliminating the i-th coordinate. This code will be denoted by $\mathrm{P}_i(C)$.

Definition 2 (Shortening) Let C be a linear code of length n and consider the subspace $C(i)$ of codewords which are zero in the coordinate $i \in \{0, \ldots, n-1\}$. Puncturing $C(i)$, which is also linear, in the coordinate i gives a linear code $\mathrm{S}_i(C)$ of length $n-1$ and this method is called the *shortening* of C in i.

Proposition 2 [9, Theorem 1.5.7] *Let C be a linear code. Then $\mathrm{S}_i(C^\perp) = \mathrm{P}_i(C)^\perp$ and $\mathrm{P}_i(C^\perp) = \mathrm{S}_i(C)^\perp$.*

Proposition 3 [9, Theorem 1.5.1] *Let C be an $[n, k, d]_q^{d^\perp}$ code and $i \in \{0, \ldots, n-1\}$.*

- *If $d > 1$, $\mathrm{P}_i(C)$ is an $[n-1, k, d']_q^{\geq d^\perp}$ code where $d' = d-1$ if C has a minimum weight codeword with nonzero i-th coordinate and $d' = d$ otherwise.*
- *When $d = 1$, $\mathrm{P}_i(C)$ is an $[n-1, k, 1]_q^{\geq d^\perp}$ code if C has no codeword of weight 1 whose nonzero entry is in coordinate i; otherwise, if $k > 1$, $\mathrm{P}_i(C)$ is an $[n-1, k-1, \geq 1]_q^{\geq d^\perp}$ code.*

Corollary 1 *Let C be an $[n, k, d]_q^{d^\perp}$ code, then $\mathrm{S}_i(C)$ is an $[n-1, k', \geq d]_q^{d'^\perp}$ code with $k' \in \{k-1, k\}$ and $d'^\perp \geq d^\perp - 1$.*

Definition 3 (Residual code) The *residual code* $\mathrm{Res}(C, c)$ with respect to a codeword $c \in C$ is the restriction of C to the zero coordinates of c.

Proposition 4 [9, Corollary 2.7.2] *Let C be an $[n,k,d]_q^{d^\perp}$ code and choose $c \in C$ with $\mathrm{wt}(c) = d$. Then $\mathrm{Res}(C,c)$ is an $\left[n-d, k-1, \geq \left\lceil \frac{d}{q} \right\rceil\right]_q$ code.*

The construction of the residual code corresponds to the $\mathrm{wt}(c)$-fold application of the puncturing operation on the support of c. Therefore, $\mathrm{Res}(C,c)$ has dual minimum distance $\geq d^\perp$.

Definition 4 (Construction Y_1) Applying the residual code construction to a minimum word of the dual code is called *construction* Y_1. It leads to a code with parameters $\left[n-d^\perp, k-d^\perp+1, \geq d\right]_q^{\geq \left\lceil \frac{d^\perp}{q} \right\rceil}$.

3 Inverting construction Y_1

In this section we present the algorithm that classifies all linear codes up to semilinear isometry given the parameters $n, k, d, d^\perp, q$. These will be fixed in the following. First of all, we define a set $T(n,k,d,d^\perp,q)$ of parity check matrices representing all nonequivalent codes with parameters $[n,k,\geq d]_q^{d^\perp}$. In general, we will call such a set a *transversal* of parity check matrices.

The method is based on an iterative approach. Starting point for this iterative method is a transversal S of those parity check matrices whose parameters are defined by construction Y_1, i.e. having parameters $[n-d^\perp, k-d^\perp+1, \geq d]_q$. This set S will also be fixed within this section. Afterward, we iteratively add $d^\perp$ further columns to these matrices and check whether this defines an element of $T(n,k,d,d^\perp,q)$ or not. Isomorph rejection is applied in all intermediate steps to guarantee a small search space.

3.1 Building a candidate set

We start with the definition of the transversal $T(n,k,d,d^\perp,q)$. Since these matrices should have a special block structure, we first have to prove that each equivalence class of $[n,k,\geq d]_q^{d^\perp}$ codes will be represented.

Lemma 1 *Let Δ be a parity check matrix of an $[n,k,d]_q^{d^\perp}$ code. The set*

$$E(\Delta, S) := \left\{ \widetilde{\Delta} = \begin{pmatrix} \Delta' & X \\ \mathbf{0_{n-d^\perp}} & \mathbf{1_{d^\perp}} \end{pmatrix} \middle| \begin{array}{l} \Delta' \in S, X \in \mathbb{F}_q^{(n-k-1)\times d^\perp}, \\ \widetilde{\Delta} \text{ equivalent to } \Delta \end{array} \right\}$$

is not empty.

Proof The dual code $C^\perp$ generated by Δ has minimum distance $d^\perp$, so we can construct an equivalent parity check matrix

$$\overline{\Delta} = \begin{pmatrix} \overline{\Delta}' & X' \\ \mathbf{0_{n-d^\perp}} & \mathbf{1_{d^\perp}} \end{pmatrix}.$$

Since S is a transversal of all parity check matrices of all linear $[n-d^\perp, k-d^\perp+1, \geq d]_q$ codes, there exists a unique $\Delta' \in S$ equivalent to $\overline{\Delta}'$. Furthermore, we can find

an invertible matrix A, a monomial matrix M and a field automorphism α with $\Delta' = A\alpha(\overline{\Delta}')M^{\mathrm{T}}$. Therefore,

$$\begin{pmatrix} A & \mathbf{0_{n-k-1}}^{\mathrm{T}} \\ \mathbf{0_{n-k-1}} & 1 \end{pmatrix} \alpha(\overline{\Delta}) \begin{pmatrix} M & \mathbf{0_{(n-d^\perp)\times d^\perp}} \\ \mathbf{0_{d^\perp\times(n-d^\perp)}} & I_{d^\perp} \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} \Delta' & A\alpha(X')M^{\mathrm{T}} \\ \mathbf{0_{n-d^\perp}} & \mathbf{1_{d^\perp}} \end{pmatrix} \in E(\Delta, S).$$

□

Using this lemma we can now define the transversal. It will contain the minimal element of $E(\Delta, S)$ for each parity check matrix Δ.

Corollary 2 *The set*

$$T(n,k,d,d^\perp,S,q) := \left\{ \min(E(\Delta,S)) \;\middle|\; \begin{array}{l} \Delta \text{ is a parity check matrix} \\ \text{of an } [n,k,\geq d]_q^{d^\perp} \text{ code} \end{array} \right\}$$

is a transversal of all $[n,k,\geq d]_q^{d^\perp}$ codes. The last $d^\perp$ columns of a matrix $\Delta \in T(n,k,d,d^\perp,S,q)$ are ordered lexicographically.

In the following we will show how to compute the transversal $T(n,k,d,d^\perp,S,q)$ by adding a further column to a transversal of parity check matrix of linear codes of length $n-1$. We have to distinguish two cases $d^\perp = 1$ and $d^\perp > 1$. These will be treated in the following two lemmata.

Lemma 2 *A matrix $\Delta \in T(n,k,d,1,S,q)$ has the following block structure $\Delta = \begin{pmatrix} \Delta' & \mathbf{0_{n-k-1}}^{\mathrm{T}} \\ \mathbf{0_{n-1}} & 1 \end{pmatrix}$ with $\Delta' \in S$.*

Proof This follows immediately from the fact that we are allowed to perform Gaussian elimination on the last column for the elements of $E(\Delta, S)$. □

This proves that we reach a candidate set $\mathcal{T}(n,k,d,1,S,q)$ containing $T(n,k,d,1,S,q)$ by extending the matrices in S by the vector $(\mathbf{0_{n-k-1}}, 1)^{\mathrm{T}}$. The next lemma provides an analogous rule for the second case.

Lemma 3 *Let $d^\perp > 1$ and $\Delta \in T(n,k,d,d^\perp,S,q)$. Then the projection $\Pi_{n-1}(\Delta)$ of Δ onto its first $n-1$ columns is an element of $T(n-1,k-1,d,d^\perp-1,S,q)$.*

Proof The projection $\Pi_{n-1}(\Delta)$ defines a parity check matrix of an $[n-1,k-1,\geq d]_q^{d^\perp-1}$ code, since it is the shortening of the code with parity check matrix Δ in the last position, see Proposition 3. Furthermore, because of the structure of Δ we know that $\Pi_{n-1}(\Delta) \in E(\Pi_{n-1}(\Delta), S)$. It remains to show that $\Pi_{n-1}(\Delta)$ is the lexicographically minimal element in $E(\Pi_{n-1}(\Delta), S)$.

Suppose it is not. Then there exist some matrix $A \in \mathrm{GL}_{n-k}(\mathbb{F}_q)$, a monomial matrix $M' \in \mathrm{GL}_{n-1}(\mathbb{F}_q)$ and a field automorphism α such that

$$E(\Pi_{n-1}(\Delta), S) \ni A\alpha(\Pi_{n-1}(\Delta))M'^{\mathrm{T}} < \Pi_{n-1}(\Delta).$$

For arbitrary $\lambda \in \mathbb{F}_q \setminus \{0\}$, the matrix $M = \begin{pmatrix} M' & \mathbf{0_{n-1}}^{\mathrm{T}} \\ \mathbf{0_{n-1}} & \lambda \end{pmatrix}$ together with α is also a semilinear isometry. Hence,

$$\begin{aligned} A\alpha(\Delta){M}^{\mathrm{T}} &= A\alpha\left((\Pi_{n-1}(\Delta), \Delta_{*,n-1})\right){M}^{\mathrm{T}} \\ &= \left(A\alpha(\Pi_{n-1}(\Delta)){M'}^{\mathrm{T}}, A\alpha(\Delta_{*,n-1})\lambda\right) < \Delta \end{aligned}$$

where we used $\Delta_{*,n-1}$ to refer to the last column of Δ.

Since the last row of $A\alpha(\Delta){M}^{\mathrm{T}}$ must have weight $d^\perp$, we conclude that the last entry in $A\alpha(\Delta_{*,n-1})$ must be nonzero. Hence, for the right choice of λ, this entry is equal to 1. For this particular λ, we know on the one hand $A\alpha(\Delta){M}^{\mathrm{T}} \in E(\Delta, S)$ and on the other hand $A\alpha(\Delta){M}^{\mathrm{T}} < \Delta$, which is a contradiction. □

Inverting this operation shows that we can build a candidate set

$$\mathcal{T}(n, k, d, d^\perp, S, q) \supseteq T(n, k, d, d^\perp, S, q)$$

via extending the matrices $\Delta \in T(n-1, k-1, d, d^\perp - 1, S, q)$ by an additional column. Since the last $d^\perp$ columns of the matrices in $T(n, k, d, d^\perp, S, q)$ appear in ascending order, it is sufficient to add a column that is larger than the last column of Δ. Furthermore, the columns must be chosen in such way that the conditions on d and $d^\perp$ are fulfilled.

3.2 Computing the transversal

The transversal $T(n, k, d, d^\perp, S, q)$ is calculated from $\mathcal{T}(n, k, d, d^\perp, S, q)$ with the help of the canonical form algorithm described in [6]. This algorithm takes a parity check matrix of a linear code and computes its canonical form, i.e. the parity check matrix of some equivalent code. The result is unique for all codes in this equivalence class.

The canonical form of a parity check matrix $\Delta \in T(n, k, d, d^\perp, S, q)$ is in general not equal to Δ. Therefore, $\mathcal{T}(n, k, d, d^\perp, S, q)$ is traversed in ascending order and we calculate the canonical form of each of these matrices. Now, the elements of $T(n, k, d, d^\perp, S, q)$ are exactly those, which produce its canonical form for the first time. This matrix and its canonical form will be stored at the same time.

The candidate sets $\mathcal{T}(n-\delta, k-\delta, d, d^\perp - \delta, S, q), 0 \leq \delta \leq d^\perp - 1$ will be constructed in a depth first search approach. This means that whenever we add a parity check matrix to the set $T(n-\delta, k-\delta, d, d^\perp - \delta, S, q)$ we will first build its successors in $\mathcal{T}(n-\delta+1, k-\delta+1, d, d^\perp - \delta + 1, S, q)$ by adding a further column. This agrees with the necessity of traversing the candidate sets in ascending order. Moreover, this approach simplifies the test on the necessary conditions on d and $d^\perp - \delta$ by a recursive update of prohibited columns.

Remark 1 Let $\widetilde{S} \subseteq S$ be the subset of parity check matrices which generate a linear code of dual minimum distance $\geq \left\lceil \frac{d^\perp}{q} \right\rceil$. Since construction Y_1 applied to a linear code with parameters $[n, k, d]_q^{d^\perp}$ produces a linear code with dual distance $\geq \left\lceil \frac{d^\perp}{q} \right\rceil$, we can similarly prove Lemma 1 for $\widetilde{S}$ and hence analogously define the transversal

$T(n,k,d,d^\perp,\widetilde{S},q)$. Therefore, if we are not interested in the transversals $T(n-\delta,k-\delta,d,d^\perp-\delta,S,q)$, $1 \leq \delta \leq d^\perp-1$ we are also allowed to start with $\widetilde{S}$ and extend those matrices by the same methods. For $\delta \neq 0$ the computed sets $T(n-\delta,k-\delta,d,d^\perp-\delta,\widetilde{S},q)$ may not define transversals. This is only guaranteed for those parameters with $\left\lceil \frac{d^\perp}{q} \right\rceil = \left\lceil \frac{d^\perp-\delta}{q} \right\rceil$.

4 Results

The online databases [8], [10] maintain lower and upper bounds on the minimum distance of a linear $[n,k,d]_q$ code for several prime powers. With the help of the algorithm developed in Section 3 we attacked small open cases.

In 16 cases – see Table 1 for the corresponding parameters – the nonexistence of a linear code meeting the upper bound in both databases is proved. These results will be described in Subsection 4.1. Furthermore, if an $[n,k,d]_q$ code does not exist, we conclude that a linear code with parameters $[n+1,k,d+1]_q$ and $[n+1,k+1,d]_q$ also cannot exist, since otherwise puncturing or shortening would give a contradiction. The puncturing ensures that the entries for the upper bounds in [8] ([10]) can only increase by one in vertical (horizontal) direction, whereas the shortening operation gives decreasing values on all diagonals from the upper-left to the lower-right.

Using construction Y_1 and the residual code, we were able to exclude even more codes. Table 2 summarizes those improvements of [8] and [10], just giving the entry which is the origin of all conclusions via puncturing and shortening. The number of deduced bounds in this regard (including the point itself) is noted in the row indexed by the corresponding reference. Sometimes, these observations lead to the same entry. In this case, it is only counted for one single parameter set $[n,k,d]_q$.

We used a computer program to generate all improvements compared to [8] in Table 2, because they were easily accessible using Magma [3]. Since we did the comparison with [10] only for these specific places by hand, there might exist other unrecognized improvements of [10].

Table 2 shows that both databases contain parameter sets n,k,q where they can give sharper bounds for the minimum distance than the other one. For 217 parameter sets we were able to improve the entries of both databases at the same time. Among these improvements 109 parameter sets establish the exact value for the minimum distance of an optimal code.

Besides these examples, we also proved that a $[37,31,6]_8$ code does not exist, which is an open case in [8], but not in [10]. Similarly, we constructed two nonequivalent $[17,11,6]_9$ codes which were unknown according to [8], but [10] already gives a construction method for one of these codes. In one case, namely for the parameters $[18,3,15]_9$, we also proved that a code with these parameters cannot exist. This is an open case in both databases [8] and [10], but on the other hand this result is already known [1].

This section ends with an application of the algorithm where we were interested in a full classification of all linear codes for given parameters in order to prove the nonexistence of an automorphism of order 7 of a putative self-dual $[72,36,16]$ code.

Given running times refer to an execution of our program written in C++ on a single core of a 2.4 GHz Intel Quad 2 processor.

	$q=2$	$q=3$			$q=4$					
n	35	22	24	28	19	21	22	27	30	39
k	10	8	14	21	8	14	16	17	21	27
d	13	10	7	5	9	6	5	8	7	9

	$q=5$			$q=7$		$q=8$
n	16	16	17	15	26	30
k	5	6	8	8	20	23
d	10	9	8	7	6	7

Table 1 Parameters of nonexistent codes

	$q=2$			$q=3$			$q=4$										
n	35	43	57	22	24	28	19	21	22	27	27	30	38	39	39	61	63
k	10	17	30	8	14	21	8	14	16	11	17	21	9	12	27	31	30
d	12	12	12	9	6	4	8	5	4	12	7	6	22	20	8	22	24
[8]	6	2	6	16	6	6	**9**	2	16	**1**	**4**	18	1	**2**	**9**	4	2
[10]	6	2	6	16	6	6	20	2	16	4	6	18	1	4	18	4	2

	$q=5$														
n	16	16	17	24	36	36	41	47	50	56	56	80	85	85	88
k	5	6	8	9	24	25	11	37	10	10	40	49	31	44	41
d	9	8	7	12	9	8	24	7	32	37	12	24	43	32	37
[8]	1	4	24	3	1	24	2	12	17	2	2	10	1	16	16
[10]	1	4	24	3	1	**22**	2	**11**	17	2	2	**0**	**0**	**0**	**0**

	$q=7$								$q=8$			
n	15	23	26	30	60	64	66	66	30	34	76	83
k	8	7	20	6	35	47	17	25	23	7	48	28
d	6	14	5	21	21	14	42	35	6	24	24	48
[8]	12	4	6	2	63	10	3	12	10	2	15	6
[10]	12	4	**4**	**0**	**0**	**1**	**0**	**0**	**4**	**0**	**0**	**0**

Table 2 Improvements compared to [8] and [10]

4.1 Improving upper bounds

This section presents several examples for the tightening of upper bounds on the minimum distances of linear codes. We use different approaches for the application of our algorithm since some of the results could not be reached directly. Another goal of this section is to give values on the cardinalities of the transversals in order to make our results verifiable.

4.1.1 The direct method

This section gives all the results which were directly achieved. This means that we computed the transversal $T(n,k,d,d^\perp,S,q)$ for all feasible values of $d^\perp$ and recognized that these are all empty. Upper bounds for $d^\perp$ are given by the database [8]. Lower bounds on $d^\perp$ can be determined by construction Y_1, see Definition 4,

n	$n-k=6$	$n-k=7$		
10	$1^0 \dots 3^0\ 4^2$			
11	$1^0 \dots 4^0\ 5^1$			1^2
12	$1^0 \dots 5^0\ 6^1$		1^1	2^{51}
13		1^1	2^6	3^{1219}
14		2^2	3^{30}	4^{7431}
15		3^7	4^{88}	5^{3797}
16		4^{13}	5^{64}	6^{261}
17		5^9	6^{17}	7^4
18		6^5	7^1	8^0
19		7^1	8^0	9^0
20		8^1	9^0	10^0
21		9^0	10^0	11^0

Table 3 Number of nonequivalent codes over $\mathbb{F}_4$ for $d \geq 6$, distinguished by $d^\perp$

taking the smallest value not leading to a contradiction with the data provided by [8].

The next two theorems serve as examples for the necessary computations. Furthermore, we will provide some corollaries which shows how the values in Table 2 can be derived.

Theorem 1 *A* $[21, 14, 6]_4$ *code does not exist.*

Proof It is easy to show that the dual distance $d^\perp$ lies in the range $9 \leq d^\perp \leq 11$. Table 3 gives the cardinalities of the intermediate transversals $T(n, k, d, d^\perp, S, q)$: An entry $d^{\perp\,x}$ states that there are x equivalence classes of $[n, k, \geq d]_q^{d^\perp}$ codes. The entries at $n - k = 6$ correspond to the initial sets S, whereas the entries in the column $n - k = 7$ are calculated by the algorithm. The entries with $d^\perp = 1$ are calculated according to Lemma 2 from the entries in the left column and the row above. By Lemma 3 we know that the entries with $d^\perp \geq 2$ are calculated from the entries written directly above. The calculation took approximately 2.5 minutes. □

Corollary 3 *A* $[19, 12, 6]_4$ *and a* $[20, 13, 6]_4$ *code is unique up to semilinear isometry. Codes with parameters* $[21 + i, 14 + i, 6]_4, i \geq 0$ *do not exist.*

Theorem 2 *A* $[16, 5, 10]_5$ *code does not exist.*

Proof For this code we know that $d^\perp = 4$. There is only a single equivalence class for codes with parameters $[12, 2, 10]_5$. This result is easily achieved. We use one representative to form the set S. Table 4 shows the cardinalities of the computed transversals. The result was achieved in 18 minutes. □

Proposition 5 *Linear codes with the following parameters do not exist:*

1. $[16 + i, 5 + i, 10]_5, i \geq 0$,
2. $[36 + i, 24 + i, 10]_5, i \geq 0$,
3. $[21, 11, 9]_5$, $[42, 12, 25]_5$,
4. $[86 + i, 55 + i, 25]_5, i \geq 0$.

n	$n-k=10$	$n-k=11$
12	$1^0\ 2^1$	
13		1^1
14		2^{39}
15		3^{1446}
16		4^0

Table 4 Number of nonequivalent codes over $\mathbb{F}_5$ for $d \geq 10$, distinguished by $d^\perp$

Proof The codes given in the first item cannot exist since iterative shortening would lead to the nonexistent $[16, 5, 10]_5$ code. The values given at the second item are derived by construction Y_1. A $[21, 11, 9]_5$ and $[42, 12, 25]_5$ code cannot exist since we tightened the upper bound on the dual distance $d^\perp \leq 9$ which leads together with construction Y_1 to a contradiction. The nonexistence of a $[42, 12, 25]_5$ code implies the nonexistence of $[86+i, 55+i, 25]_5$ codes, $i \geq 0$, again by construction Y_1. □

The next theorem gives more results using similar computations.

Theorem 3 *Linear codes with the following parameters do not exist:*

- $[35, 10, 13]_2$
- $[28, 21, 5]_3$
- $[19, 8, 9]_4$, $[22, 16, 5]_4$
- $[16, 6, 9]_5$, $[17, 8, 8]_5$
- $[15, 8, 7]_7$

4.1.2 Combining the classification for the code and its dual

In this section we prove the nonexistence of a $[22, 8, 10]_3^{d^\perp}$ code. The result is achieved by combining the nonexistence results for linear codes with parameters $[22, 8, \geq 10]_3^4$, $[22, 8, \geq 10]_3^5$ and $[22, 14, \geq 6]_3^{10}$.

Theorem 4 *A $[22, 8, 10]_3$ code does not exist.*

Proof Suppose there is a linear code with parameters $[22, 8, 10]_3$. By [8] we know that its dual distance must be either equal to 4, 5 or 6. Table 5, which was computed in 4 days, proves that there are no $[22, 8, 10]_3^{\leq 5}$ codes. We were not able to finish a classification of the transversal $T(22, 8, 10, 6, S, 3)$.

Furthermore, the nonexistence of a $[22, 8, 10]_3^6$ code can be shown by the fact that the transversal $T(22, 14, 6, 10, S', 3)$ is empty, see Table 6. This result was achieved within a minute. Note that the numbers of nonequivalent codes for $n \leq 21$ is not equal to the total number since we only started from the transversal $\widetilde{S'}$ of linear codes having dual distance at least 4, following Remark 1. □

4.1.3 Partial results for smaller parameter sets

Like for the case above, we were able to classify linear codes with parameters $[23, 13, 7]_3^8$. But we were not able to finish the computation for $d^\perp = 9$. In contrast

n	$n-k=13$		$n-k=14$		
15		1^2			
16	1^0	2^{12}			
17	$1^0 \dots 2^0$	3^{18}			1^{12}
18	$1^0 \dots 3^0$	4^7		1^{18}	$2^?$
19			1^7	2^{134679}	$3^?$
20			2^{1926}	$3^{4273434}$	$4^?$
21			3^{61}	4^{4641}	$5^?$
22			4^0	5^0	$6^?$
23			5^0	6^0	7^0 (cannot exist by dual upper bound)

Table 5 Number of nonequivalent codes over $\mathbb{F}_3$ for $d \geq 10$, distinguished by $d^\perp$

n	$n-k=7$		$n-k=8$
12	$1^? \dots 3^?$	4^1	
13			$1^{1+?}$
14			$2^{40+?}$
15			$3^{768+?}$
16			$4^{4851+?}$
17			$5^{1817+?}$
18			$6^{179+?}$
19			$7^{48+?}$
20			$8^{3+?}$
21			$9^{0+?}$
22			10^0

Table 6 Number of nonequivalent codes over $\mathbb{F}_3$ for $d \geq 6$, distinguished by $d^\perp$

to the result above, we were also not able to exclude the existence of a $[23, 13, 7]_3^9$ code by the dual approach.

Nevertheless, the excluded cases for an $[n, k, \geq d]_q^\delta$ code give a lower bound $l \leq d^\perp$ which also implies that the dual distance of an $[n+i, k+i, \geq d]$ code has to be greater than $l+i$. The result is now due to the fact that the upper bound $u(n, k, q)$ on the minimum distance of a linear $[n, k]$ code does not always increase by one if n is increased by one. Hence, as soon as this lower bound exceeds its upper bound $u(n+i, n-k, q)$ we have disproved the existence of an $[n+i, k+i, \geq d]$ code.

In the example above, we can use our partial result of the nonexistence of a $[23, 13, 7]_3^8$ code in order to prove the nonexistence of a $[24, 14, 7]_3$ code. By Theorem 4 we know that the minimum distance of a $[24, 10]_3$ code is at most 9. Hence, we can us the fact that $u(23, 10, 3) = u(24, 10, 3) = 9$ and hence we only have to investigate the transversal $T(24, 14, 7, 9, S, 3)$ which is empty since $T(23, 13, 7, 8, S, 3)$ is already empty.

This idea was also applied in several other cases, whenever we were not able to exclude the existence of an $[n, k, \geq d]^{d^\perp}$ code for all cases of $d^\perp$.

Theorem 5 *Linear $[n, k, d]_q$ codes with parameters listed in the first column of Table 7 do not exist.*

excluded parameters	feasible $d^\perp$	by nonexistence of	missing $d^\perp$
$[24, 14, 7]_3$	9	$[23, 13, \geq 7]_3^{\leq 8}$	9
$[27, 17, 8]_4$	$9 \ldots 13$	$[21, 11, \geq 8]_4^{\leq 7}$	8
$[30, 21, 7]_4$	$13 \ldots 16$	$[22, 13, \geq 7]_4^{\leq 8}$	9, 10
$[26, 20, 6]_7$	18	$[16, 10, \geq 6]_7^{8}$	9
$[30, 23, 7]_8$	21	$[16, 9, \geq 7]_8^{7}$	8

Table 7 Results from partial classifications for $d^\perp$

Parameters of D_1			$\lvert T(10, k, d, d^\perp, 8)\rvert$
k	d	$d^\perp$	
3	8	4	1
4	4	4	81717
4	5	4	1854753
4	6	4	490382
5	4	4	61487808
5	5	4	3742898
5	5	5	3014997
Total			70672556

Table 8 Number of nonequivalent candidates for D_1

Proof We proved the nonexistence for linear codes with parameters given in the third column of Table 7. This implies that a linear $[n, k, d]_q^{d^\perp}$ code could not exist for all of its feasible values $d^\perp$ listed in the second column. □

The next theorem gives one further conclusion made by the classification of smaller codes.

Theorem 6 *A* $[39, 27, 9]_4$ *code does not exist.*

Proof By [8] a $[39, 27, 9]_4$ code has dual distance ≤ 21. The nonexistence of a $[19, 8, \geq 9]_4$ code (also proved by our method, see Table 1) implies that $d^\perp \geq 21$. The code derived by construction Y_1 has parameters $[18, 7, \geq 9]_4^{\geq 6}$. By our algorithm we prove that no such code can exist and hence there is no $[39, 27, 9]_4$ code. □

4.2 The automorphism group of a self-dual $[72, 36, 16]_2$ code does not contain Z_7

In [7] it is shown that a self-dual $[72, 36, 16]_2$ code has no automorphism of order 7. In this computational proof we were forced to classify all linear codes D_1 of length 10 over $\mathbb{F}_8$ whose minimum distance and dual minimum distance is greater than or equal to 4. Table 8 gives the result if one uses the symmetry of interchanging the role of D_1 and its dual $D_1^\perp$.

References

1. Barnabei M., Searby D., Zucchini C.: On small (k, q)-arcs in planes of order q^2, J. Combin. Theory Ser. A, 24, 241–246 (1978).
2. Betten, A., Braun, M., Fripertinger, H., Kerber, A., Kohnert, A., Wassermann, A.: Error-Correcting Linear Codes. Springer-Verlag, Berlin (2006)
3. Bosma W., Cannon J. J., Fieker C., Steel A. (eds.): Handbook of Magma Functions, Edition 2.16 (2010).
4. Bouyukliev, I.G., Jacobsson, E.: Results on binary linear codes with minimum distance 8 and 10. IEEE Trans. Inform. Theory, 57 6089–6093 (2011).
5. Edel, Y., Bierbrauer, J.: Inverting construction Y_1. IEEE Trans. Inform. Theory, 44, 1993 (1998)
6. Feulner, T.: The automorphism groups of linear codes and canonical representatives of their semilinear isometry classes. Adv. Math. Commun., 3, 363–383 (2009)
7. Feulner, T., Nebe, G.: The automorphism group of a self-dual binary $[72, 36, 16]$ code does not contain Z_7 or $Z_3 \times Z_3$. arXiv:abs/1110.6012 (2011)
8. Grassl, M.: Bounds on the minimum distance of linear codes and quantum codes. http://www.codetables.de (2012). Accessed 22 February 2012
9. Huffman, W. C., Pless V.: Fundamentals of Error-Correcting Codes. Cambridge University Press, Cambridge, 2003
10. Schmid W., Schürer R.: MinT: a database for optimal net parameters. In: Niederreiter, H. and Talay, D. (eds.) Monte Carlo and Quasi-Monte Carlo Methods 2004, pp. 457–469, Springer, Heidelberg (2006). http://mint.sbg.ac.at/. Accessed 22 February 2012