

On the Classification of Self-Dual Additive Codes over GF(9)

Lars Eirik Danielsen

Department of Informatics, University of Bergen, N-5020 Bergen, Norway

Email: larsed@ii.uib.no

Abstract—Additive codes over GF(9) that are self-dual with respect to the Hermitian trace inner product have previously been classified up to length 8. In this paper, all codes of length 9 and 10 are classified, using a new algorithm that combines two graph representations of codes. First, the search space is reduced by the fact that every self-dual additive code can be mapped to a weighted graph. Then a different graph is described that transforms the problem of code equivalence into a problem of graph isomorphism.

I. INTRODUCTION

An *additive* code, \mathcal{C} , over GF(9) of length n is a GF(3)-linear subgroup of GF(9) n . \mathcal{C} contains 3^k codewords for some $0 \leq k \leq 2n$, and can be defined by a $k \times n$ *generator matrix*, with entries from GF(9), whose rows span \mathcal{C} additively. \mathcal{C} is called an $(n, 3^k)$ code. We denote GF(9) = $\{0, 1, \omega, \omega^2, \dots, \omega^7\}$, where $\omega^2 = \omega + 1$ and $\omega^4 = -1$. *Conjugation* of $x \in \text{GF}(9)$ is defined by $\bar{x} = x^3$. The *trace map*, $\text{Tr} : \text{GF}(9) \rightarrow \text{GF}(3)$, is defined by $\text{Tr}(x) = x + \bar{x}$. The *Hermitian trace inner product* of two vectors $\mathbf{u}, \mathbf{v} \in \text{GF}(9)^n$ is given by $(\mathbf{u}, \mathbf{v}) = \omega^2(\mathbf{u} \cdot \bar{\mathbf{v}} - \bar{\mathbf{u}} \cdot \mathbf{v}) = \text{Tr}(\omega^2 \mathbf{u} \cdot \bar{\mathbf{v}})$, where multiplication by ω^2 ensures that the product takes values in GF(3) [1]. We define the *dual* of the code \mathcal{C} with respect to the Hermitian trace inner product, $\mathcal{C}^\perp = \{\mathbf{u} \in \text{GF}(9)^n \mid (\mathbf{u}, \mathbf{c}) = 0 \text{ for all } \mathbf{c} \in \mathcal{C}\}$. If $\mathcal{C} = \mathcal{C}^\perp$, then \mathcal{C} is *self-dual* and must be an $(n, 3^n)$ code. The *Hamming weight* of \mathbf{u} , denoted $\text{wt}(\mathbf{u})$, is the number of non-zero components of \mathbf{u} . The *Hamming distance* between \mathbf{u} and \mathbf{v} is $\text{wt}(\mathbf{u} - \mathbf{v})$. The *minimum distance* of the code \mathcal{C} is the minimal Hamming distance between any two distinct codewords of \mathcal{C} . Since \mathcal{C} is an additive code, the minimum distance is also given by the smallest non-zero weight of any codeword in \mathcal{C} . A code with minimum distance d is called an $(n, 3^k, d)$ code. The *weight distribution* of the code \mathcal{C} is the sequence (A_0, A_1, \dots, A_n) , where A_i is the number of codewords of weight i . The *weight enumerator* of \mathcal{C} is the polynomial $W(y) = \sum_{i=0}^n A_i y^i$. It follows from the *Singleton bound* [2] that any self-dual additive code must satisfy $d \leq \lfloor \frac{n}{2} \rfloor + 1$. \mathcal{C} is called *extremal* if it attains the minimum distance d given by this bound, and *near-extremal* if it has minimum distance $d - 1$. If a code has highest possible minimum distance for the given length, it is called *optimal*.

Two self-dual additive codes over GF(9) are *equivalent* if the codewords of one can be mapped onto the codewords of the other by a transformation that preserves the properties of the code, i.e., weight enumerator, additivity, and self-duality. It is known [2] that this group of transformations

is $\text{Sp}_2(3) \wr \text{Sym}(n)$, i.e., permutations of the coordinates combined with operations from the *symplectic group* $\text{Sp}_2(3)$ applied independently to each coordinate.¹ Let an element $a + b\omega \in \text{GF}(9)$ be represented as $\begin{pmatrix} a \\ b \end{pmatrix} \in \text{GF}(3)^2$. We can then premultiply this element by a 2×2 matrix. The group $\text{Sp}_2(3) = \left\langle \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \right\rangle$ has order 24 and contains all 2×2 matrices with elements from GF(3) and determinant one. The order of $\text{Sp}_2(3) \wr \text{Sym}(n)$ is $24^n n!$, and hence this is the total number of maps that take a self-dual additive code over GF(9) to an equivalent code. By translating the action of $\text{Sp}_2(3)$ on $\begin{pmatrix} a \\ b \end{pmatrix}$ into operations on elements $c = a + b\omega \in \text{GF}(9)$, we find that the operations we can apply to all elements in a coordinate of a code are $c \mapsto xc$ if $x^4 = 1$, or $c \mapsto x\bar{c}$ if $x^4 = -1$, given $x \in \text{GF}(9)$, and $a + b\omega \mapsto a + yb + b\omega$, for $y \in \text{GF}(3)$.

A transformation that maps \mathcal{C} to itself is called an *automorphism* of \mathcal{C} . All automorphisms of \mathcal{C} make up the *automorphism group* of \mathcal{C} , denoted $\text{Aut}(\mathcal{C})$. The number of distinct codes equivalent to \mathcal{C} is then given by $\frac{24^n n!}{|\text{Aut}(\mathcal{C})|}$. The *equivalence class* of \mathcal{C} contains all distinct codes that are equivalent to \mathcal{C} . By adding the sizes of all equivalence classes of codes of length n , we find the total number of distinct codes of length n , denoted T_n . The number T_n is also given by a *mass formula* [1], [3]:

$$T_n = \prod_{i=1}^n (3^i + 1) = \sum_{j=1}^{t_n} \frac{24^n n!}{|\text{Aut}(\mathcal{C}_j)|}, \quad (1)$$

where t_n is the number of equivalence classes of codes of length n , and \mathcal{C}_j is a representative from each equivalence class. The smallest possible automorphism group, called the *trivial automorphism group*, of a self-dual additive code over GF(9) is $\{I, -I\}$, i.e., it consists of global multiplication of coordinates by 1 or -1 . By assuming that all codes of length n have a trivial automorphism group, we obtain from the mass formula a lower bound on t_n :

$$t_n \geq \left\lceil \frac{2 \prod_{i=1}^n (3^i + 1)}{24^n n!} \right\rceil. \quad (2)$$

Note that when n is large, most codes have a trivial automorphism group, so the tightness of the bound increases with n . As we will see in Section V, for $n = 10$, 80% of all codes have a

¹Global conjugation of all coordinates will also preserve the properties of the code. This *weak equivalence* [1] is not considered in this paper.

trivial automorphism group, and the bound (2) underestimates t_{10} by just 19%.

Any linear code over $\text{GF}(9)$ which is self-dual with respect to the Hermitian inner product, $(\mathbf{u}, \mathbf{v}) = \mathbf{u} \cdot \bar{\mathbf{v}}$, is also a self-dual additive code with respect to the Hermitian trace inner product. The operations that map a self-dual linear code to an equivalent code are more restrictive than for additive codes, since $\text{GF}(9)$ -linearity must now be preserved. Only coordinate permutations and multiplication of single coordinates by $x \in \text{GF}(9)$ where $x^4 = 1$ is allowed. It follows that only additive codes that satisfy certain constraints can be equivalent to linear codes.²

Trace-Hermitian self-dual additive codes over $\text{GF}(q)$ exist for $q = r^2$, where r is a prime power. The first case, $\text{GF}(4)$, has been studied in detail, in particular since an application to quantum error-correction was discovered [4]. The next case, $\text{GF}(9)$, has received less attention, although these codes have similar application in quantum error-correction [2].

In Section II we briefly review the connection between self-dual additive codes and weighted graphs. The main contribution of this paper is a new algorithm for checking equivalence of self-dual additive codes over $\text{GF}(9)$, described in Section III, which is a generalization of a known algorithm for linear codes [5]. Combining this algorithm with the weighted graph representation, and some other optimizations, enables us to classify all codes of length up to 10 in Section IV. In particular, all near-extremal codes of length 9 and 10 are classified for the first time. We also find the smallest codes with trivial automorphism group.

II. CODES AND WEIGHTED GRAPHS

An m -weighted graph is a triple $G = (V, E, W)$, where V is a set of vertices, $E \subseteq V \times V$ is a set of edges, and W is a set of weights from $\text{GF}(m)$, such that each edge has an associated non-zero weight. In an unweighted graph, which can simply be described by a pair $G = (V, E)$, all edges have weight one. A graph with n vertices can be represented by an $n \times n$ adjacency matrix Γ , where the element $\Gamma_{i,j} = W(\{i, j\})$ if $\{i, j\} \in E$, and $\Gamma_{i,j} = 0$ otherwise. A loop-free undirected graph has a symmetric adjacency matrix where all diagonal elements are 0. In a directed graph, edges are ordered pairs, and the adjacency matrix is not necessarily symmetric. In a colored graph, the set of vertices has been partitioned into disjoint subsets, and each subset has been assigned a color. Two graphs $G = (V, E)$ and $G' = (V, E')$ are isomorphic if and only if there exists a permutation π of V such that $\{u, v\} \in E \iff \{\pi(u), \pi(v)\} \in E'$. We also require that edge weights are preserved, i.e., $W(\{u, v\}) = W(\{\pi(u), \pi(v)\})$. In a colored graph, we require the permutation to be color preserving. The automorphism group of a graph is the set of vertex permutations that map the graph to itself. A path is a sequence of vertices, (v_1, v_2, \dots, v_i) , such that $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_{i-1}, v_i\} \in E$. A graph is connected if there is a path from any vertex to any other vertex in the graph.

²The correspondence between the additive codes classified in this paper and linear codes will be considered in future work.

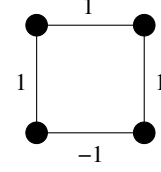


Fig. 1. Graph Representation of $(4, 3^4, 3)$ Code

If an additive code over $\text{GF}(9)$ has a generator matrix of the form $C = \Gamma + \omega I$, where I is the identity matrix, ω is a primitive element of $\text{GF}(9)$, and Γ is the adjacency matrix of a loop-free undirected 3-weighted graph, we say that the generator matrix is in standard form. A generator matrix in standard form must generate a code that is self-dual with respect to the Hermitian trace inner product, since it has full rank over $\text{GF}(3)$ and $CC^T = \Gamma^2 + \Gamma - I$ only contains entries from $\text{GF}(3)$, and hence the traces of all elements of $w^2 CC^T$ will be zero.

It is known that every self-dual additive code is equivalent to a code with a generator matrix in standard form [6], [7]. Write the generator matrix $C = A + \omega B$ as an $n \times 2n$ matrix $(A \mid B)$ with elements from $\text{GF}(3)$. If B has full rank, we simply perform the basis change $B^{-1}(A \mid B) = (\Gamma \mid I)$. Elements on the diagonal of Γ can be set to zero by operations $a + b\omega \mapsto a + yb + b\omega$, for $y \in \text{GF}(3)$. In the case where B has rank $k < n$, we can assume, after a basis change, that the first k rows and columns of B form a $k \times k$ invertible matrix. By the operation $c \mapsto \omega \bar{c}$, for $c = a + b\omega$, corresponding to the symplectic matrix $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$, we can replace column a_i by $-b_i$ and b_i by a_i . In this way, we “swap” the $n - k$ last columns of A and B . It can be shown, from the self-duality of the code, that the new matrix B must have full rank [8], [9], and the standard form is then obtained as above. It also follows from self-duality that the obtained Γ will always be a symmetric matrix.

As an example, consider the $(4, 3^4, 3)$ code generated by C which by the described method is transformed into the standard form generator matrix C' , corresponding to the weighted graph depicted in Fig. 1:

$$C = \begin{pmatrix} 1 & 0 & 1 & \omega^2 \\ \omega & 0 & \omega & \omega^3 \\ 0 & 1 & \omega^2 & 1 \\ 0 & \omega & \omega^3 & \omega \end{pmatrix} C' = \begin{pmatrix} \omega & -1 & 0 & 1 \\ -1 & \omega & 1 & 0 \\ 0 & 1 & \omega & 1 \\ 1 & 0 & 1 & \omega \end{pmatrix}$$

It is known that two self-dual additive codes over $\text{GF}(4)$ are equivalent if and only if their corresponding graphs are related by a sequence of graph operations called local complementations (LC) [8], [10] and a permutation of the vertices. Equivalence classes of self-dual additive codes over $\text{GF}(9)$ can also be represented as orbits of weighted graphs with respect to a generalization of LC [11]. We have previously used LC to classify self-dual additive codes over $\text{GF}(4)$ of length up to 12 [12], but were only able to reach length 8 for self-dual additive codes over $\text{GF}(9)$ [9]. The main problem

is that the size of the LC orbits of weighted graphs quickly get unmanageable as the number of vertices increase. We must therefore use another method for checking code equivalence, which is described in the next section. This algorithm uses a graph representation of self-dual additive codes over $\text{GF}(9)$ that is not related to the representation described in this section, and does not require the input to be in standard form. However, the weighted graph representation will still be very useful in reducing our search space.

III. EQUIVALENCE GRAPHS

To check whether two self-dual additive codes over $\text{GF}(9)$ are equivalent, we modify a known algorithm used for checking equivalence of linear codes [5]. The idea is to map a code to an unweighted, directed, colored *equivalence graph* such that the automorphism groups of the code and the equivalence graph coincide.

An important component of the algorithm is to find a suitable *coordinate graph*. For self-dual additive codes over $\text{GF}(9)$, we need to construct a graph G on eight vertices, labeled with the non-zero elements from $\text{GF}(9)$, whose automorphism group is $\text{Sp}_2(3)$. This graph, shown as a subgraph of Fig. 2, was found by adding directed edges $(\sigma 1, \sigma \omega)$ for all $\sigma \in \text{Sp}_2(3)$. This ensures that $\text{Sp}_2(3) \subseteq \text{Aut}(G)$. We then verify that $|\text{Aut}(G)| = 24$ which implies that $\text{Aut}(G) = \text{Sp}_2(3)$.

Fig. 3 shows examples of coordinate graphs for some other families of codes over $\text{GF}(9)$. In the original algorithm for checking equivalence of linear codes [5], the coordinate graph shown in Fig. 3a would be used. This graph has an automorphism group of size eight, corresponding to the fact that multiplication of a coordinate by any non-zero element from $\text{GF}(9)$ preserves linearity. The more restrictive coordinate graph for Hermitian self-dual linear codes over $\text{GF}(9)$ is shown in Fig. 3b. This graph has an automorphism group of size four, since only multiplication by $x \in \text{GF}(9)$ where $x^4 = 1$ is permitted in this case.

To construct the equivalence graph of a code, we add n copies of the coordinate graph, each copy representing one coordinate of the code. We then need to find a set of codewords of some weights that generate the code. First, we check if the set of all codewords of minimum weight d suffices. If it does not, we add all codewords of weight $d + 1$, then all codewords of weight $d + 2$, etc, until we have a set of codewords that generate the code. For each codeword c_i in the resulting set, we add a *codeword vertex* v_i to the equivalence graph. Let the codeword vertices have one color, and the other vertices have a different color. Edges are added between v_i and the coordinate graphs according to the non-zero coordinates of the codeword c_i , e.g., if c_i has ω in coordinate j , then there is an edge between v_i and the vertex labeled ω in the j th coordinate graph. As an example, Fig. 2 shows the case where $c_1 = (\omega 1 \cdots 1)$. The resulting equivalence graph is finally *canonicalized*, i.e., relabeled, but with coloring preserved, using the *nauty* software [13]. If two graphs are isomorphic, their canonical representations are guaranteed to be the same.

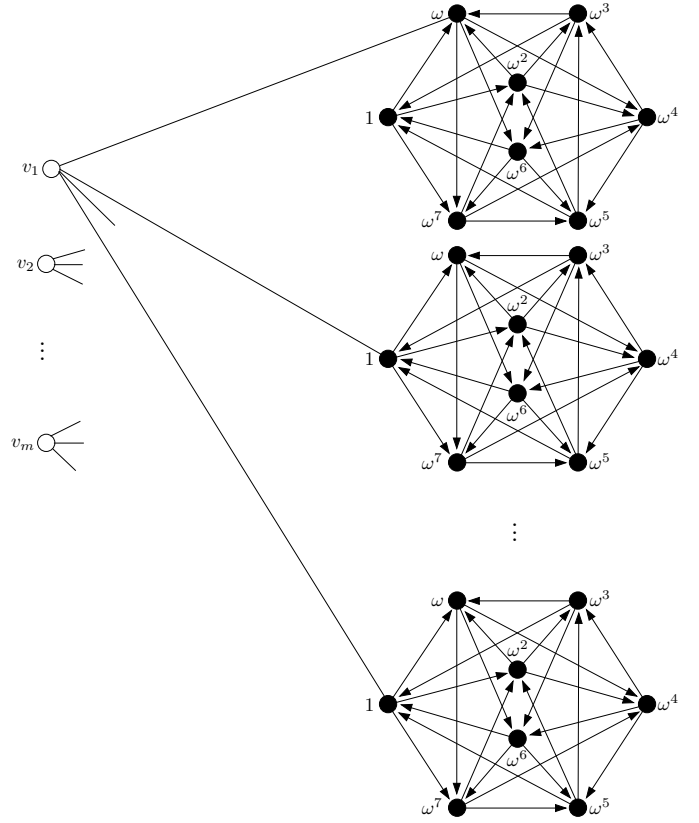


Fig. 2. Example of Equivalence Graph

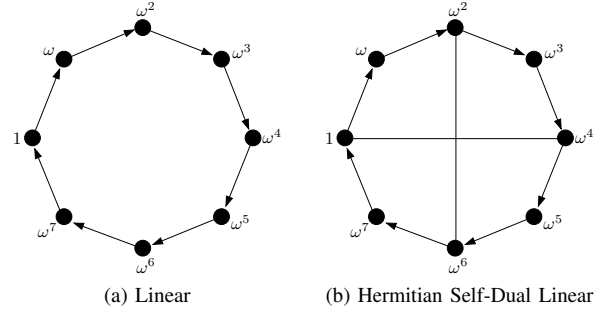


Fig. 3. Coordinate Graphs for Other Codes over $\text{GF}(9)$

Applying a canonical permutation to the vertices of an equivalence graph corresponds to permuting the coordinates of the corresponding code, applying elements from $\text{Sp}_2(3)$ to each coordinate, and sorting the codewords c_i in some canonical order. If two codes are equivalent, their canonical equivalence graphs will be identical. Furthermore, the automorphism group of a code is equivalent to the automorphism group of its equivalence graph. This follows from the fact that any automorphism of the equivalence graph must be one out of $24^n n!$ possibilities, i.e., the $n!$ permutations of the n coordinate subgraphs, and the 24 automorphisms of each coordinate subgraph. No other automorphisms are possible. In particular, permuting the codeword vertices will never be an automorphism, since all codewords must be distinct. Since it

is known [2] that coordinate permutations and $\text{Sp}_2(3)$ applied to the coordinates of a code preserve its weight enumerator, additivity, and self-duality, this must also be true for any automorphism of the equivalence graph.

IV. CLASSIFICATION

Since every weighted graph corresponds to a self-dual additive code, and every self-dual additive code, up to equivalence, can be represented as a weighted graph, we only need to consider 3-weighted graphs to classify all self-dual additive codes over $\text{GF}(9)$. Permuting vertices of a graph corresponds to permuting coordinates of the associated code, which means that we only need to consider these graphs up to isomorphism. Moreover, we can restrict our study to connected graphs, since a disconnected graph represents a *decomposable* code. A code is decomposable if it can be written as the *direct sum* of two smaller codes. For example, let \mathcal{C} be an $(n, 3^n, d)$ code and \mathcal{C}' an $(n', 3^{n'}, d')$ code. The direct sum, $\mathcal{C} \oplus \mathcal{C}' = \{u||v \mid u \in \mathcal{C}, v \in \mathcal{C}'\}$, where $||$ means concatenation, is an $(n+n', 3^{n+n'}, \min\{d, d'\})$ code. All decomposable codes of length n can be generated easily once all indecomposable codes of length less than n are known.

To classify codes of length n , we could take all non-isomorphic connected 3-weighted graphs on n vertices, map the corresponding codes to equivalence graphs, and canonize these. All duplicates would then be removed to obtain one representative code from each equivalence class. However, a much smaller set of graphs is obtained by taking all possible *lengthenings* [14] of all codes of length $n-1$. A generator matrix in standard form can be lengthened in $3^{n-1} - 1$ ways, by adding a vertex to the corresponding graph, and connecting it to all possible combinations of at least one of the old vertices, using all possible combinations of edge weights. This corresponds to adding a new row $\mathbf{r} \in \text{GF}(3)^n$ and column \mathbf{r}^T to the adjacency matrix, with zero in the last coordinate. Only half of the lengthenings need to be considered, since adding the row $-\mathbf{r}$ is equivalent to adding \mathbf{r} . This can be verified by multiplying row and column n in the corresponding generator matrix by -1 . It has been shown [9], using the theory of local complementation of weighted graphs, that the set of $i_{n-1} \frac{3^{n-1}-1}{2}$ codes obtained by lengthening one representative from each of the i_{n-1} equivalence classes of indecomposable codes of length $n-1$ must contain at least one representative from each equivalence class of the indecomposable codes of length n .

Removing possible isomorphisms from the set of lengthened graphs, using *nauty* [13], speeds up our classification significantly. A set of non-isomorphic graphs that have already been processed, as large as memory resources permit, can even be stored between iterations, and new graphs can be checked for isomorphism against this set. For each graph that is not excluded by such an isomorphism check, the corresponding code must be mapped to an equivalence graph which is canonized and compared with all previously generated codes, which are stored in memory. Since the equivalence graphs will be large, typically containing thousands of vertices for

TABLE I
NUMBER OF INDECOMPOSABLE (i_n) AND TOTAL NUMBER (t_n) OF SELF-DUAL ADDITIVE CODES OVER $\text{GF}(9)$ OF LENGTH n

n	1	2	3	4	5	6	7	8	9	10
i_n	1	1	1	3	5	21	73	659	17 589	2 803 404
t_n	1	2	3	7	13	39	121	817	18 525	2 822 779

TABLE II
NUMBER OF INDECOMPOSABLE SELF-DUAL ADDITIVE CODES OVER $\text{GF}(9)$ OF LENGTH n AND MINIMUM DISTANCE d

$d \backslash n$	2	3	4	5	6	7	8	9	10
2	1	1	2	4	15	51	388	6240	418 088
3			1	1	5	20	194	6975	893 422
4					1	2	77	4370	1 487 316
5								4	4577
6									1
All	1	1	3	5	21	73	659	17 589	2 803 404

$n = 10$, we map the equivalence graph to a canonical generator matrix by taking the first n linearly independent codewords corresponding to codeword vertices in their canonical ordering. This generator matrix can further be mapped to a canonical standard form, as described in Section II, which means that only $\binom{n}{2}$ ternary symbols need to be stored for each code.

The special form of a generator matrix in standard form makes it easy to find all codewords of low weight, which is necessary to construct the equivalence graph. If \mathcal{C} is generated by $C = \Gamma + \omega I$, then any codeword formed by taking $\text{GF}(3)$ -linear combinations of i rows of C must have weight at least i . This means that we can find all codewords of weight i by only considering combinations of at most i rows of C .

V. CONCLUSION

Using the described approach, we have classified all codes of length up to 10. Table I gives the values of i_n , the number of inequivalent indecomposable codes of length n , and the values of t_n , the total number of inequivalent codes of length n . Table II lists the numbers of indecomposable codes by length and minimum distance. In Table III, we count the distinct weight enumerators. There are obviously too many codes of length 9 and 10 to list all of them here, so an on-line database containing one representative from each equivalence class has been made available at <http://www.ii.uib.no/~larsed/nonbinary/>. The extremal codes of length 9 and 10 have been described before [9], but the classification of the 4370 near-extremal $(9, 3^9, 4)$ codes and 4577 near-extremal $(10, 3^{10}, 5)$ codes is new.

That our classification is correct has been verified by the mass formula (1). This required us to also calculate the sizes of the automorphism groups of all decomposable codes, which was simplified by the observation that for a code $\mathcal{C} = k_1 \mathcal{C}_1 \oplus \dots \oplus k_m \mathcal{C}_m$, where $k_j \mathcal{C}_j = \bigoplus_{i=1}^{k_j} \mathcal{C}_j$, $|\text{Aut}(\mathcal{C})| = \prod_{i=1}^m k_i! |\text{Aut}(\mathcal{C}_i)|^{k_i}$.

TABLE III
NUMBER OF DISTINCT WEIGHT ENUMERATORS OF INDECOMPOSABLE
CODES OF LENGTH n AND MINIMUM DISTANCE d

$d \backslash n$	2	3	4	5	6	7	8	9	10
2	1	1	2	4	14	42	202	1021	8396
3			1	1	3	9	33	170	1133
4					1	1	9	25	345
5								1	10
6									1
All	1	1	3	5	18	52	244	1217	9885

TABLE IV
NUMBER OF CODES OF LENGTH n AND MINIMUM DISTANCE d WITH
TRIVIAL AUTOMORPHISM GROUP

$d \backslash n$	≤ 7	8	9	10
3		32	4518	832 878
4		3	3056	1 419 861
5				3795
All	0	35	7574	2 256 534

We find that the smallest codes with trivial automorphism group are 35 codes of length 8.³ Generator matrices for these codes can be obtained from <http://www.iu.uib.no/~larsed/nonbinary/>. Table IV gives the numbers of codes with trivial automorphism group by length and minimum distance. It is easy to show that codes with minimum distance $d \leq 2$ will always have nontrivial automorphisms. For $d = 1$, we can assume that the first column of a standard form generator matrix is $(\omega 0 \dots 0)^T$. Then $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \in \text{Sp}_2(3)$ applied to the first coordinate of the code is an automorphism of order 3. For $d = 2$, we can assume that the first column of a standard form generator matrix is $(\omega 10 \dots 0)^T$, then $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \in \text{Sp}_2(3)$ applied to the first coordinate and $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \in \text{Sp}_2(3)$ applied to the second coordinate of the code has the same effect as adding the first row of the generator matrix to the second row, and is hence an automorphism of order 3.

According to the mass formula bound (2), the total number of codes of length 11 is $t_{11} \geq 1\,592\,385\,579$, which makes a complete classification infeasible with our computational resources. Running our algorithm on a typical desktop computer, the classification of codes of length n was completed in less than five minutes for $n \leq 8$, about two hours for $n = 9$, and about a week for $n = 10$. Most of this time is spent canonizing the equivalence graphs with *nauty*, and far more time is used on codes with large automorphism groups than on codes with trivial or small automorphism groups. This means that our previous classification algorithm [9], using local complementation, might still be useful in some cases, since we observe that graphs corresponding to codes with

large automorphism groups typically have small LC orbits. For instance, we could speed up our classification algorithm by not only removing isomorphisms from the set of lengthened codes, but also generating and storing a limited number of LC orbit members of each graph, and checking new graphs for isomorphism against this set.

By lengthening an $(n, 3^n, d)$ code, we obtain an $(n + 1, 3^{n+1}, d')$ code, where $d' \leq d + 1$ [14]. It follows that given a classification of all codes of length n and minimum distance at least d , we can classify all codes of length $n + 1$ and minimum distance at least $d + 1$. There are no $(11, 3^{11}, 6)$ codes, but by lengthening the 1 491 894 $(10, 3^{10}, d)$ codes for $d \geq 4$, we would obtain all optimal $(11, 3^{11}, 5)$ codes. Preliminary results show that there are several million inequivalent such codes. If this classification can be completed, it should also be feasible, and of greater interest, to find all all optimal $(12, 3^{12}, 6)$ codes.

ACKNOWLEDGEMENT

The author would like to thank an anonymous reviewer for providing useful suggestions and corrections. This research was supported by the Research Council of Norway.

REFERENCES

- [1] G. Nebe, E. M. Rains, and N. J. A. Sloane, *Self-dual codes and invariant theory*, ser. Algorithms and Computation in Mathematics. Berlin: Springer-Verlag, 2006, vol. 17.
- [2] E. M. Rains, "Nonbinary quantum codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1827–1832, 1999.
- [3] G. Höhn, "Self-dual codes over the Kleinian four group," *Math. Ann.*, vol. 327, no. 2, pp. 227–255, 2003.
- [4] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, "Quantum error correction via codes over GF(4)," *IEEE Trans. Inform. Theory*, vol. 44, no. 4, pp. 1369–1387, 1998.
- [5] P. R. J. Östergård, "Classifying subspaces of Hamming spaces," *Des. Codes Cryptogr.*, vol. 27, no. 3, pp. 297–305, 2002.
- [6] M. Grassl, A. Klappenecker, and M. Rötteler, "Graphs, quadratic forms, and quantum codes," in *Proc. IEEE Int. Symp. Inform. Theory*, 2002, p. 45.
- [7] D. Schlingemann, "Stabilizer codes can be realized as graph codes," *Quantum Inf. Comput.*, vol. 2, no. 4, pp. 307–323, 2002.
- [8] M. Van den Nest, J. Dehaene, and B. De Moor, "Graphical description of the action of local Clifford transformations on graph states," *Phys. Rev. A*, vol. 69, no. 2, p. 022316, 2004.
- [9] L. E. Danielsen, "Graph-based classification of self-dual additive codes over finite fields," *Adv. Math. Commun.*, vol. 3, no. 4, pp. 329–348, 2009.
- [10] A. Bouchet, "Graphic presentations of isotropic systems," *J. Combin. Theory Ser. B*, vol. 45, no. 1, pp. 58–76, 1988.
- [11] M. Bahramgiri and S. Beigi, "Graph states under the action of local Clifford group in non-binary case," 2006, preprint. [Online]. Available: <http://arxiv.org/abs/quant-ph/0610267>
- [12] L. E. Danielsen and M. G. Parker, "On the classification of all self-dual additive codes over GF(4) of length up to 12," *J. Combin. Theory Ser. A*, vol. 113, no. 7, pp. 1351–1367, 2006.
- [13] B. D. McKay, *nauty User's Guide, Version 2.4*, Nov. 2009. [Online]. Available: <http://cs.anu.edu.au/~bdm/nauty/nug.pdf>
- [14] P. Gaborit, W. C. Huffman, J.-L. Kim, and V. Pless, "On additive GF(4) codes," in *Codes and Association Schemes*, ser. DIMACS Ser. Discrete Math. Theoret. Comput. Sci., vol. 56. Providence, RI: Amer. Math. Soc., 2001, pp. 135–149.

³Automorphism group sizes were not calculated in the previous classification of codes of length 8 [9].