



# A new algorithm for equivalence of cyclic codes and its applications

Nuh Aydin<sup>1</sup> · R. Oliver Vandenberg<sup>1</sup>

Received: 15 May 2020 / Revised: 24 June 2021 / Accepted: 27 August 2021 /  
Published online: 6 September 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

## Abstract

Cyclic codes are among the most important families of codes in coding theory for both theoretical and practical reasons. Despite their prominence and intensive research on cyclic codes for over a half century, there are still open problems related to cyclic codes. In this work, we use recent results on the equivalence of cyclic codes to create a more efficient algorithm to partition cyclic codes by equivalence based on cyclotomic cosets. This algorithm is then implemented to carry out computer searches for both cyclic codes and quasi-cyclic (QC) codes with good parameters. We also generalize these results to repeated-root cases. We have found several new linear codes that are cyclic or QC as an application of the new approach, as well as more desirable constructions for linear codes with best known parameters. With the additional new codes obtained through standard constructions, we have found a total of 14 new linear codes.

**Keywords** Best known linear codes · Cyclic codes · Cyclotomic cosets · Equivalence of codes · Search algorithms for linear codes

**Mathematics Subject Classification** 94B15 · 94B60

## 1 Introduction

We study the problem of determining equivalence between cyclic codes in this work. There are several reasons that motivate this study. The class of cyclic codes is one of the oldest families in coding theory that have been studied for over 6 years [17, 18]. They are important for both theoretical and practical purposes and they establish

---

✉ Nuh Aydin  
aydinn@kenyon.edu  
R. Oliver Vandenberg  
vandenberg1@kenyon.edu

<sup>1</sup> Department of Mathematics and Statistics, Kenyon College, Gambier, OH, USA 43022

a fundamental link between coding theory and algebra. There has been intensive research on codes over rings in the past few decades. Whenever a new type of ring is introduced to coding theory, one of the very first things researchers do is to study cyclic codes over that ring. In addition to the fact that both cyclic codes and code equivalence are fundamental topics in coding theory, there are additional benefits that may be obtained from new results on this question as explained below.

A linear code  $C$  over a finite field  $\mathbb{F}_q$  is a vector subspace of  $\mathbb{F}_q^n$ . Each code has a length ( $n$ ), dimension ( $k$ ), and minimum Hamming distance ( $d$ ). Constructing codes with optimal parameters is one of the most important and challenging problems in coding theory. This optimization problem can be formulated in a few different ways. For example, given the length  $n$  and the dimension  $k$  over a certain finite field  $\mathbb{F}_q$ , we look for the largest possible value  $d_q(n, k)$  of the minimum distance. Theoretical upper bounds are available for  $d_q(n, k)$ , and codes whose parameters attain an upper bound are called optimal codes, sometimes  $d$ -optimal codes. The online database [12] gives information about best known linear codes (BKLC) over small finite fields  $\mathbb{F}_q$ ,  $q \leq 9$ , including lower and upper bounds on  $d_q(n, k)$ . Lower bounds are usually obtained by explicit constructions. We observe from the database that in most cases there are gaps between lower and upper bounds on  $d_q(n, k)$ . In general, optimal codes are known only when either  $k$  or  $n - k$  is small.

Computer searches are often used to find linear codes with best known parameters, however there are inherent limitations to the computational approach. First, determining the minimum distance of a linear code is computationally intractable [23] so it takes significant time to find the minimum distance of a single code when the dimension is large. Second, for a given length and dimension, the number  $\frac{(q^n - 1)(q^n - q) \cdots (q^n - q^{k-1})}{(q^k - 1)(q^k - q) \cdots (q^k - q^{k-1})}$  of linear codes is very large and grows quickly. Hence, exhaustive computer searches on linear codes is not feasible. Therefore, we focus on specific classes of codes with rich mathematical structures that are known to contain many codes with good parameters. One such family of codes is the class of quasi-twisted (QT) codes that contains cyclic, constacyclic and quasi-cyclic (QC) codes as special cases. Hundreds of record breaking codes have been obtained in the last few decades by computer searches within these classes (see [4–6, 10, 22] for a few examples) all of which are generalizations of cyclic codes. Our work in this paper shows that new results on equivalence of cyclic codes can be useful for computational purposes.

Another area of research that has been receiving much attention recently is code-based cryptography due to its promise for the age of quantum computers. One specific problem of great interest in the field is to reduce the key size in the McEliece cryptosystem [15], one of the earliest examples of public key cryptosystem—as old as the widely used RSA cryptosystem—that did not become widely used due to its large key size. However, given the promise of the McEliece system for post-quantum cryptography there has been a renewed interest on the subject. Cyclic codes and their various generalizations, such as QC codes, are being considered in an effort to reduce the key size in the McEliece cryptosystem (see [7] for a recent example). Code equivalence is of fundamental importance in code-based cryptography which provides another motivation for this work.

## 2 Basic definitions

A linear code  $C$  is called cyclic if whenever  $c = (c_0, c_1, \dots, c_{n-1})$  is a codeword, then so is its cyclic shift  $\pi(c) = (c_{n-1}, c_0, c_1, \dots, c_{n-2})$ . If we represent a vector  $c = (c_0, c_1, \dots, c_{n-1})$  by the polynomial  $c(x) = c_0 + c_1x + \dots + c_{n-1}x^{n-1}$ , then we obtain a vector space isomorphism between  $\mathbb{F}_q^n$  and the set of all polynomials over  $\mathbb{F}_q$  of degree less than  $n$ . In this representation, the cyclic shift of  $c$  in  $\mathbb{F}_q^n$  corresponds to the multiplication of  $c(x)$  by  $x$  in the quotient ring  $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$ . It is well known that cyclic codes of length  $n$  over  $\mathbb{F}_q$  are precisely the ideals of the ring  $\mathbb{F}_q[x]/\langle x^n - 1 \rangle$  which is a principal ideal ring (a ring in which every ideal is generated by a single element) and there is a one-to-one correspondence between cyclic codes of length  $n$  over  $\mathbb{F}_q$  and divisors of  $x^n - 1$  over  $\mathbb{F}_q$ . In general, a cyclic code  $C$  has many generators. Among all generators of  $C$ , there is a generator polynomial  $g(x)$  that is uniquely determined by the following two conditions (i)  $g(x)$  is monic (ii)  $g(x)$  is a non-zero polynomial of smallest degree in  $C$ . We write  $C = \langle g(x) \rangle$ . This unique generator must divide  $x^n - 1$ , and it is called the (standard) generator of  $C$ . Hence, we can write  $x^n - 1 = g(x)h(x)$  and call  $h(x)$  the check polynomial of  $C$ . Moreover, any other generator of  $C$  is of the form  $g(x)f(x)$  where  $\gcd(f(x), h(x)) = 1$  [3]. A cyclic code may be defined either by its generator polynomial or its check polynomial.

Given the generator polynomial  $g(x) = g_0 + g_1x + \dots + g_r x^r$  of a cyclic code  $C$ , we obtain a generator matrix for  $C$  as a circulant matrix of the form

$$\begin{bmatrix} g_0 & g_1 & \dots & g_r & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_r & 0 & \dots & 0 \\ \dots & & & & & & & \\ 0 & \dots & 0 & g_0 & g_1 & \dots & g_r \end{bmatrix}.$$

where each row is the cyclic shift of the row above it. The dimension of  $C$  is  $n - \deg(g(x))$ .

A fundamental notion in the study of cyclic codes is that of cyclotomic coset, a subset of  $\mathbb{Z}_n$ .

**Definition 1** Let  $\gcd(n, q) = 1$ . For any  $s \in \mathbb{Z}_n$ , the  $q$ -cyclotomic coset of  $n$  containing  $s$  is the set  $S_s = \{sq^j \pmod n : j = 0, 1, 2, \dots\}$

Assuming  $\gcd(n, q) = 1$ , there is a one-to-one correspondence between the irreducible factors of  $x^n - 1$  and cyclotomic cosets mod  $n$ . Therefore there is a correspondence between cyclic codes and unions of cyclotomic cosets when  $\gcd(n, q) = 1$ . This correspondence is obtained via the  $n$ th roots of unity. Let  $\alpha \in \mathbb{F}_{q^t}$  be a primitive  $n$ th root of unity over  $\mathbb{F}_q$ . Then,  $x^n - 1 = \prod_{i=0}^{n-1} (x - \alpha^i)$  in  $\mathbb{F}_{q^t}[x]$ . Each cyclotomic coset  $S_s = \{s, sq, sq^2, \dots, sq^{r-1}\} \pmod n$  corresponds to an irreducible divisor  $p(x)$  of  $x^n - 1$  over  $\mathbb{F}_q$  by  $p(x) = \prod_{i \in S_s} (x - \alpha^i)$ . We denote the cyclotomic coset corresponding with a given polynomial  $g$  by  $S_g$ .

Equivalence of codes is a central concept for this work. Two linear codes are called equivalent if one can be obtained from the other by any combination of the following transformations

1. A permutation of the coordinates.
2. Multiplication of elements in a fixed position by a non-zero scalar in  $\mathbb{F}_q$ .
3. Applying a field automorphism of  $\mathbb{F}_q$  to each component of the vectors.

If only (1) is used, then the codes are called permutation equivalent. This is a very important special case. We can summarize all of these conditions in the following way.

**Definition 2** Two linear codes  $C_1, C_2 \subseteq \mathbb{F}_q^n$  are equivalent if there exists a monomial matrix  $M$  and an automorphism  $\phi$  over  $\mathbb{F}_q$  such that  $C_1 = C_2 M \phi$ .

It is well known that equivalent codes have the same parameters. This has important implications for computer searches because a search algorithm need not examine any code that is equivalent to codes which have been already searched. This work generalizes our previous results in [2] on the equivalence of cyclic codes and their generalization to constacyclic codes. Further, it includes a new algorithm for partitioning cyclic codes by equivalence and a search algorithm that only checks the minimum distance of codes that are not equivalent to any of the codes already examined.

### 3 Extending results on equivalence of cyclic codes to repeated root case

Under the assumption of  $\gcd(n, q) = 1$ , the following theorems are proven in [2].

**Theorem 1** [2] *Let  $g_1(x)$  and  $g_2(x)$  be the standard generators of cyclic codes of length  $n$  over  $\mathbb{F}_q$  and assume  $\gcd(e, n) = 1$ . Then the isometry*

$$\Phi : \mathbb{F}_q[x]/\langle x^n - 1 \rangle \longrightarrow \mathbb{F}_q[x]/\langle x^n - 1 \rangle$$

given by

$$x \mapsto x^e \pmod{x^n - 1}$$

has the property  $g_2(x) = \phi(g_1(x))$  if and only if the map  $\phi : S_{g_1} \mapsto S_{g_2}$  given by  $\phi(z) = e^{-1}z \pmod{n}$ , where  $e^{-1}$  is the multiplicative inverse of  $e \pmod{n}$ , is a bijection.

**Theorem 2** [2] *Let  $g_1(x)$  be the standard generator of a cyclic code of length  $n$  over  $\mathbb{F}_q$  where  $\gcd(n, q) = 1$ , and let  $\delta = \alpha^{-b}$  where  $\alpha$  is a primitive  $n^{\text{th}}$  root of unity, such that  $n$  divides  $b \cdot \deg(g_1(x)) \cdot (q - 1)$ . Let  $K$  be an extension field of  $\mathbb{F}_q$  that contains  $\delta$ . Then the isometry  $\Phi : K[x]/\langle x^n - 1 \rangle \mapsto K[x]/\langle x^n - 1 \rangle$  defined by*

$\Phi(f(x)) = f(\delta x) \pmod{(x^n - 1)}$  has the property that  $\Phi(g_1(x)) \in \mathbb{F}_q[x]$  and generates a cyclic code of length  $n$  over  $\mathbb{F}_q$  if and only if the map  $\phi : \mathbb{Z}_n \mapsto \mathbb{Z}_n$  defined by  $\phi(z) = z + b \pmod n$  is a bijection such that  $\phi(S_{g_1}) = S_{\Phi(g_1)}$ .

These theorems in their current form require that  $x^n - 1$  does not have repeated roots. This was assumed largely because there was not a construction for cyclotomic cosets in repeated-root cases. We now propose such a construction. First write the code length  $n$  over a field with characteristic  $q$  as  $n = n_q q^i$  where  $\gcd(q, n_q) = 1$ . Next we find the cyclotomic cosets mod  $n_q$ .

Define a function  $P$  which takes cyclotomic cosets to polynomials. Let  $\alpha$  be an  $n_q^{\text{th}}$  root of unity, and  $S$  be a cyclotomic coset mod  $n$ . We define  $P(S) = \prod_{i \in S} (x - \alpha^i)$ . Then we use a multiset to describe unions where if an irreducible factor of  $x^n - 1$  appears multiple times in a divisor, then the roots of that factor appear multiple times in the multiset. This means a multiset  $MS$  is a union of not necessarily distinct cyclotomic cosets  $S_1, S_2, \dots, S_k$  and it corresponds to the polynomial  $P(MS) = P(S_1) \cdot P(S_2) \cdots P(S_k)$ .

For example, let us consider binary cyclic codes of length 14. Then  $n_q = 7$ . So the cyclotomic cosets are  $S_0 = \{0\}$ ,  $S_1 = \{1, 2, 4\}$  and  $S_3 = \{3, 5, 6\}$ . Now the following are two possible multisets describing a union that corresponds to a divisor of  $x^{14} - 1$ :  $MS_1 = \{1, 2, 4, 1, 2, 4\}$  and  $MS_2 = \{3, 5, 6, 3, 5, 6\}$ . The corresponding codes to these multisets are equivalent via a map between the sets of the form given in Theorem 1 with  $e^{-1} = 5$ . These correspond to the polynomials  $P(MS_1) = P(S_1)P(S_1) = x^6 + x^4 + 1$  and  $P(MS_2) = P(S_3)P(S_3) = x^6 + x^2 + 1$  which indeed generate equivalent codes.

Because the proofs of the Theorems 1 and 2 given in [2] do not rely on the fact that the cyclotomic cosets have distinct elements, and use the same function to take cyclotomic cosets to polynomials, both theorems also apply to this repeated-root construction as well. In generalizing the results in Theorems 1 and 2 to the repeated root case, one needs to replace the condition  $\gcd(e, n) = 1$  by  $\gcd(e, n_q) = 1$ .

Moreover, this method can be generalized to apply to constacyclic codes too. This is because  $x^n - a|x^m - 1$ , where  $r$  is the order of  $a$  in the multiplicative group of  $\mathbb{F}_q^*$ . The irreducible factors of  $x^n - a$  over  $\mathbb{F}_q$  correspond to a subset of cyclotomic cosets mod  $nr$ . See [3] for more on this.

### 4 A new equivalence testing algorithm for cyclic codes

The theorems in the previous section are the basis of an algorithm that we developed and implemented for testing cyclic code equivalence. This algorithm takes as input the alphabet of the codes, the length of the codes, and two cyclotomic cosets (or unions of cyclotomic cosets) that define each cyclic code. The output of the algorithm is a Boolean variable (called map in the pseudocode below) which indicates whether the two cyclic codes are equivalent.

---

**Algorithm 1** Equivalence Testing Algorithm for Cyclic Codes
 

---

```

1: Set alphabet, field, and code length  $(q^m, F, n)$ 
2: Enter cyclotomic cosets (gen1, gen2) of two cyclic codes.
3: map=false;
4: i=1;
5: while map eq 0 and i lt n do
6:   if GCD(i,n) eq 1 then
7:     b=0;
8:     while map eq 0 and b lt n do
9:       count=0;
10:      for j=1 to #gen1 do
11:        if Multiplicity(gen1,gen1[j]) eq Multiplicity(gen2,((gen2[j]*i+b) mod n)) then
12:          count=count+1;
13:        end if; end for;
14:        if count eq #gen1 then
15:          map=true;
16:        end if;
17:        b=b+1; end while;
18:      end if;
19:    i=i+1; end while;

```

---

A few important remarks about this algorithm are in order. First, a cyclotomic coset does not uniquely determine an irreducible factor of  $x^n - 1$ . It depends on the choice of the primitive  $n$ th root of unity. Nevertheless, different choices for the primitive  $n$ th root of unity will lead to equivalent codes. Secondly, since this algorithm is based on theorems that give sufficient conditions for two cyclic codes to be equivalent, when the algorithm returns that the two cyclic codes are equivalent, the output is definitely correct, and in this case the map between cyclotomic cosets is determined by the variables  $i$  and  $b$ . However, when the output is false, it does not guarantee that the codes are not equivalent (except for the binary case. See section 7). In that sense, it is similar to probabilistic primality testing algorithms that are true-biased. Despite this limitation, this algorithm is still very useful for our purposes as explained in the next section.

## 5 Computational complexity of cyclic equivalence test algorithm

We consider the complexity of checking if there is a bijective map of the form  $ex + b$  between the cyclotomic cosets corresponding with cyclic codes of length  $n$ . It is easy to see that the complexity is  $O(n^3)$  since all possible combinations of  $e, b < n$  can be tested (a total of  $\leq n^2$  choices) to see if for each element of one cyclotomic coset is in the other cyclotomic coset at cost proportional to  $n$ . This shows for cyclic codes testing equivalence using their cyclotomic cosets is a polynomial time algorithm in the length of the codes.

Determining whether two arbitrary linear codes are equivalent is much more challenging ([19–21]). It has been shown that testing code equivalence generally is equivalent to the graph isomorphism problem [16]. For cyclic codes, using cyclotomic cosets is a much more efficient test. It is particularly superior at large

dimensions. This is because for a cyclic code of dimension  $k$ , the size of the cyclotomic coset is  $n - k$ , so it runs faster at higher dimensions, unlike other algorithms such as Magma [14] implementation (available using either `IsEquivalent` or `IsIsomorphic` command), which run slower. In comparing our algorithm with Magma's, it should be noted that (a) our algorithm is specifically for cyclic codes whereas Magma's algorithm is for general linear codes (b) given two cyclic codes, if our algorithm concludes that they are equivalent, then the result is certainly correct. However, when it does not make that conclusion, the codes may still be equivalent.

Given these restrictions, it is not entirely fair to compare our algorithm with Magma's in general. However, for our purposes the comparison is justifiable because Magma does not offer any algorithm specifically for cyclic codes (which is an easier case than general linear codes) and there are many cases where Magma's algorithm get stuck or takes so much time or memory that it is not useful. In fact, it was these shortcomings of Magma's algorithm that prompted us to come up with a more efficient algorithm for cyclic codes in the first place. Therefore, there are clear benefits and advantages of the new algorithm over existing algorithms of Magma, and a comparison is still relevant.

The following table shows a comparison of Magma's `IsEquivalent` command and our algorithm on different generator polynomials in terms of CPU time and memory usage. In each case the generators were for codes of the same length and dimension, which were not necessarily equivalent. All of the test cases were executed on the online Magma calculator<sup>1</sup> for consistency.

$q$	$n$	$k$	equiv	CycCoset		IsEquivalent	
				Time	Memory	Time	Memory
2	135	111	True	.04s	32MB	6.04s	32MB
3	80	61	True	.01s	32MB	> 30s	*
3	80	19	True	.01s	32MB	.1s	32MB
5	72	14	False	.14s	32MB	.2s	32MB
5	72	18	False	.12s	32MB	> 83s	*
7	18	12	True	.01s	32MB	.34s	32MB
7	36	28	False	.02s	32MB	3.490s	96.16MB
7	72	62	False	.03s	32MB	**	Unfinished
7	72	10	False	.14s	32MB	.2s	32MB

\* On these runs the Magma calculator's memory limit of 353MB was exceeded, and the time at which it was exceeded is given.

\*\* On this run the 120s time limit was exceeded.

<sup>1</sup> <http://magma.maths.usyd.edu.au/calc/>

## 6 Application to a search algorithm

Our new algorithm that partitions cyclic codes of a given length and dimension into equivalence classes is purely based on combinations of cyclotomic cosets. It generates all possible cyclic codes using the unions of cyclotomic cosets. Each new set is compared against all sets that have been found up to that point. If it is equivalent to a previous one then it will be discarded. Each cyclotomic coset that is not found to be equivalent to any other is saved and the corresponding generator polynomial is obtained. Computational evidence from testing many different cases shows that this new algorithm is both faster and less likely to get stuck than the Magma's algorithm.

This partitioning algorithm has applications to both constacyclic searches and quasi-twisted (QT) searches. Earlier versions of the constacyclic search algorithm (all versions before [2], such as [4]) found the (standard) generators of all constacyclic codes of a given length one by one and computed the minimum distance for each code. Now, by first partitioning the constacyclic codes into equivalence classes, we only need to compute the minimum distance of one code from each equivalence class. Since computing the minimum distance of a linear code is computationally expensive [23], the new approach makes the search faster. Moreover, the new method of checking code equivalence based on cyclotomic cosets is much faster than Magma's algorithm. This further speeds up the search over the previous algorithm [2]. Taking advantage of these optimizations, our search produced 3 new cyclic codes with better parameters than previously best known linear codes [12]. It is highly desirable but not very common (in fact, it is often impossible) that a best known code for a given parameter set has the cyclic structure. Our search also produced many cyclic codes that have the same parameters as currently BKLCs in [12]. The constructions of currently BKLCs in [12] are frequently indirect, involving a long chain of steps. It would therefore be desirable to obtain them as cyclic codes whenever possible.

One of our main motivations for this work was to speed up the search algorithm ASR. First introduced in [3], ASR is a search algorithm for a particular type of 1-generator QT codes. It is based on Theorem 3.2 in [3]. It was then used in many subsequent works (e.g. [1, 5, 6, 9, 10]) and produced dozens of new linear codes. We recently introduced a generalization of the ASR algorithm for searching QT codes [2] which made the original version of the ASR algorithm [3] more general by making use of the concept of code equivalence. In [2], we implemented the generalized algorithm using Magma software and found a number of new linear codes. We used Magma's `IsEquivalent` command to test equivalence of cyclic codes, which did not always work. For example, it experienced a bug in  $GF(3)$ . This bug was recently fixed, but still this command is often very slow and uses much more memory than the algorithm proposed here. Magma's algorithm is for checking equivalence between any two linear codes. We are not aware of another version that is specifically for cyclic codes. The algorithm we propose here is specifically for cyclic codes and it uses our partitioning algorithm as a much faster way of obtaining the generators to be fed into the generalized ASR algorithm.



We now present our partition algorithm which uses Algorithm 1 above. It takes as input the code length  $n$ , and the size  $q^m$  of the finite field (code alphabet) which need not be relatively prime. As the output, it creates a list whose elements are coset representatives for each equivalence class of cyclic codes of length  $n$  over  $GF(q^m)$ .

---

### Algorithm 2 Partition Algorithm for Cyclic Codes

---

```

1: Set alphabet, field, and code length  $(q^m, F, n)$ 
2: module= $n$ ; c1= $0$ ;
3: while (module mod  $q$ ) eq  $0$  do
4:   module=module div  $q$ ;
5: c1=c1+1; end while;
6: cycreprs= $q^{c1}$ ;
7: Generate all cyclotomic cosets (ocoset).
8: totalnum=(cycreprs+1)#ocosets - 1;
9: partit=[[ ]];
10: for i=1 to totalnum do
11:   Generate  $i^{th}$  coset (poscos)
12:   Compare poscos to all cosets in partit using Algorithm 1.
13:   if it is not equivalent to any coset in partit then
14:     Store poscos in partit
15:   end if; end for;

```

---

Through an implementation of this algorithm we have been able to search for QC codes over  $GF(3)$  which yielded 4 new record breaking codes. We were originally not able to carry out this search in [2] due to a bug in Magma. We point out that this algorithm can be modified for constacyclic codes and QT codes using the constacyclic cyclotomic coset construction given in [3].

## 7 A limitation to the partition

We conjectured in [2] (Conjecture 3) that all cyclic codes that are equivalent are equivalent through an affine map of the form  $\phi(x) = ex + b$  on cyclotomic cosets. We have found a few counterexamples to the conjecture for lengths which are multiples of 8 over  $GF(3)$  and  $GF(5)$ . Consider the length 8 over  $GF(3)$ . The cyclotomic cosets are  $\{0\}$ ,  $\{1, 3\}$ ,  $\{2, 6\}$ ,  $\{4\}$ , and  $\{5, 7\}$ . Two possible unions of cyclotomic cosets are  $\{0, 1, 3, 4\}$  and  $\{1, 2, 3, 6\}$ . These turn out to correspond to equivalent codes even though no map of the form  $ex + b$  exists between these cyclotomic cosets (verified by exhaustive testing).

The following matrix, obtained by Magma command, maps one code (its generator matrix) to the other.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

We observed that the structure of this matrix is similar to the counterexamples we have found for larger lengths and we are still looking for a pattern between the corresponding cyclotomic cosets.

The implication of this limitation for our search is that in some cases the partitions may have too many equivalence classes. These searches will still be exhaustive, but because of these counterexamples they may also have some redundancy in certain cases. Still, since the counterexamples seem to be rare, the method proposed is a more efficient partitioning algorithm.

In the important special case of binary codes, the only notion of equivalence is permutation equivalence. More generally, it is shown in [11] that for cyclic codes of length  $n$  over  $\mathbb{F}_q$ , the notions of permutation equivalence and monomial equivalence coincide if and only if  $\gcd(n, q-1) = 1$ . Over the binary field all equivalent codes we have observed have a map of the form  $x \rightarrow x^e$  between their generator polynomials. It turns out that this is a consequence of a result given in [8] (Theorem 5.81, page 142). Therefore, for binary codes the output of Algorithm 1 is certain (i.e., it is a full test) and it is a faster alternative to Magma's algorithm.

## 8 New linear codes

As a result of an implementation of this algorithm, we have found 3 new ("record breaking", i.e., codes with better parameters than currently BKLCs reported in [12]) linear codes that are cyclic. We like to point out that it is rare for a record breaking code to be cyclic. Of the hundreds of new linear codes that have been discovered by computer searches over the last few decades, very few of them are cyclic. Since cyclic codes have a special structure that makes their implementation easier, it is desirable to have BKLCs as cyclic codes. This work presents one of the rare occasions where newly discovered linear codes are cyclic. Their parameters and generators are as follows:

- [146, 122, 9]<sub>3</sub> cyclic code with generator
 
$$x^{24} + 2x^{23} + x^{22} + 2x^{21} + 2x^{20} + x^{19} + x^{16} + 2x^{14} + x^{13} + x^{11} + 2x^{10} + x^8 + x^5 + 2x^4 + 2x^3 + x^2 + 2x + 1$$
- [146, 121, 9]<sub>3</sub> cyclic code with generator
 
$$x^{25} + x^{24} + 2x^{23} + x^{22} + 2x^{20} + 2x^{19} + x^{17} + 2x^{16} + 2x^{15} + 2x^{14} + 2x^{13} + x^{12} + x^{11} + x^{10} + x^9 + 2x^8 + x^6 + x^5 + 2x^3 + x^2 + 2x + 2$$
- [78, 63, 8]<sub>5</sub> cyclic code with generator

$$x^{15} + 3x^{14} + x^{13} + x^{11} + 3x^9 + x^8 + 3x^7 + 4x^6 + 2x^5 + 2x^3 + x^2 + 4x + 4$$

We have also found 4 new ternary QC codes that are record breaking among linear codes. Their parameters and generators are listed below. They are all 1-generator QC codes of index 3 ( $\ell = 3$ ) with a generator of the form  $(g(x), g(x)f_2(x), g(x)f_3(x))$ . To represent a polynomial in a compact way, we just list the coefficients in ascending order of the terms. For example, the sequence [2021] represents the polynomial  $2 + 2x^2 + x^3$ .

$[n, k, d]_q$		Generator Polynomials
[60, 19, 22] <sub>3</sub>	g	=[21]
	$f_2$	=[2200021200110200111]
	$f_3$	=[0012002212221102101]
[72, 22, 26] <sub>3</sub>	g	=[101]
	$f_2$	=[1122220222021210022212]
	$f_3$	=[1220021122022111]
[72, 21, 27] <sub>3</sub>	g	=[1221]
	$f_2$	=[002100021111200121202]
	$f_3$	=[200112121120102020202]
[72, 18, 29] <sub>3</sub>	g	=[1120221]
	$f_2$	=[010110000212001001]
	$f_3$	=[1210221200221001]
New QC Codes		

Moreover, additional new codes are obtained from these codes using the standard constructions of shortening, extending, or puncturing a given linear code. The additional new codes we obtained have the following parameters: [59, 18, 22], [70, 19, 27], [71, 20, 27], [71, 21, 26], [73, 21, 27], [73, 22, 27], [74, 22, 27]. With these, the total number of new linear codes we have obtained in this work is 14.

### 8.1 Cyclic codes with the same parameters as BKLCs

As noted above, in many cases the constructions of currently best known linear codes in [12] are indirect that may involve multiple steps and manipulations. For example, the construction of the binary [171, 134, 10]<sub>2</sub> code in [12] involves 4 steps starting with a BCH code of length 257, then puncturing it, then extending the resulting code, then shortening the resulting code at many positions and finally taking a subcode. The construction of the [71, 55, 8]<sub>5</sub> code in [12] requires 7 similar steps. It would therefore be desirable to obtain BKLCs as cyclic codes whenever possible. Our search produced many cyclic codes that have the same parameters as currently BKLCs where the constructions of BKLCs in [12] are indirect with many steps. We present a subset of such codes here.

- [171, 134, 10]<sub>2</sub> cyclic code with generator

- [10010000001000110111101100010000001001]
- [129, 30, 38]<sub>2</sub> cyclic code with generator  
[1000100011101001001011100010001]
- [126, 100, 8]<sub>2</sub> cyclic code with generator  
[11110111110111111011011011]
- [56, 42, 6]<sub>3</sub> cyclic code with generator  
[101201221201011]
- [80, 46, 14]<sub>3</sub> cyclic code with generator  
[12022102001111011100122212122201111]
- [80, 60, 8]<sub>3</sub> cyclic code with generator  
[201120202000121220211]
- [60, 53, 4]<sub>3</sub> cyclic code with generator  
[21001011]
- [51, 26, 14]<sub>4</sub> cyclic code with generator  
[1aa<sup>2</sup>a<sup>2</sup>aaa<sup>2</sup>11a<sup>2</sup>1a1101aaa0aa<sup>2</sup>11a<sup>2</sup>1]
- [170, 154, 6]<sub>4</sub> cyclic code with generator  
[1a<sup>2</sup>1a11aa10a<sup>2</sup>11]
- [71, 55, 8]<sub>5</sub> cyclic code with generator  
[12420401032243421]
- [120, 114, 4]<sub>5</sub> cyclic code with generator  
[4424141]
- [48, 24, 17]<sub>7</sub> cyclic code with generator  
[5025253510542120656461511]
- [42, 32, 6]<sub>7</sub> cyclic code with generator  
[66541233451]

## References

1. Ackerman, R., Aydin, N.: New quinary linear codes from quasi-twisted codes and their duals. *Appl. Math. Lett.* **24**(4), 512–515 (2011)
2. Aydin, N., Lambrinos, J.G., VandenBerg, R.O.: On equivalence of cyclic Codes, generalization of a quasi-twisted search algorithm, and new linear codes. *Des. Codes Cryptogr.* **87**, 2199–2212 (2019). <https://doi.org/10.1007/s10623-019-00613-0>
3. Aydin, N., Siap, I., Ray-Chaudhuri, D.K.: The structure of 1-generator quasi-twisted codes and new linear codes. *Des. Codes Cryptogr.* **24**(3), 313–326 (2001)
4. Aydin, N., Murphree, J.: New linear codes from constacyclic codes. *J. Frankl. Inst.* **351**(3), 1691–1699 (2014)
5. Aydin, N., Connolly, N., Murphree, J.: New binary linear codes from quasi-cyclic codes and an augmentation algorithm. *Appl. Algebra Eng. Commun. Comput.* **28**(4), 339–350 (2017)
6. Aydin, N., Connolly, N., Grassl, M.: Some results on the structure of constacyclic codes and new linear codes over GF(7) from quasi-twisted codes. *Adv. Math. Commun.* **11**(1), 245–258 (2017)
7. Baldi, M., Barenghi, A., Chiaraluce, F., Pelosi, G., Santini, O.: LEDAkem: A Post-quantum Key Encapsulation Mechanism Based on QC-LDPC Codes, pp. 3–24. Springer, Cham (2018)
8. Berlekamp, E.R.: *Algebraic Coding Theory*. Aegean Park Press, Walnut Creek (1984)
9. Daskalov, R., Hristov, P.: New quasi-twisted degenerate ternary linear codes. *IEEE Trans. Inform. Theory* **49**(9), 2259–2263 (2003)

10. Daskalov, R., Hristov, P., Metodieva, E.: New minimum distance bounds for linear codes over  $GF(5)$ . *Discrete Math.* **275**(1), 97–110 (2004)
11. Dobson, E.: Monomial isomorphisms of cyclic codes. *Des. Codes Cryptogr.* **76**(2), 257–267 (2015)
12. M. Grassl: Code Tables: Bounds on the parameters of codes, online, <http://www.codetables.de/>
13. Huffman, W.C., Pless, P.: *Fundamentals of Error-Correcting Codes*. Cambridge University Press, Cambridge (2003)
14. Magma computer algebra system, online, <http://magma.maths.usyd.edu.au/>
15. McEliece, R.J.: A public-key cryptosystem based on algebraic coding theory. Tech. Rep. 44, Jet Propulsion Lab. (1978)
16. Petrank, E., Roth, R.M.: Is code equivalence easy to decide? *IEEE Trans. Inform. Theory* **43**(5), 1602–1604 (1997)
17. Prange, E.: Cyclic error-correcting codes in two symbols. Technical Report TN-57-103, Air Force. Cambridge Research Center, Cambridge (1957)
18. Prange, E.: Some cyclic error-correcting codes with simple decoding algorithm. Technical Report TN-58-156, Air Force. Cambridge Research Center, Cambridge (1958)
19. Sendrier, N.: Finding the permutation between equivalent linear codes: the support splitting algorithm. *IEEE Trans. Inform. Theory* **46**(4), 1193–1203 (2000)
20. Sendrier, N., Simos, D.E.: How easy is code equivalence over  $\mathbb{F}_q$ ? In: *Proceeding of International Workshop on Coding Theory and Crypt.*, Bergen, Norway (2013)
21. Sendrier, N., Simos, D.E.: The hardness of code equivalence over  $\mathbb{F}_q$  and its application to code-based cryptography. In: Gaborit, P. (ed.) *Post-Quantum Cryptography*, Springer Lecture Notes in Computer Science 7932, pp. 203–216. France, Limoges (2013)
22. Siap, I., Aydin, N., Ray-Chaudhuri, D.K.: New ternary quasi-cyclic codes with improved minimum distances. *IEEE Trans. Inform. Theory* **46**(4), 1554–1558 (2000)
23. Vardy, A.: The intractability of computing the minimum distance of a code. *IEEE Trans. Inform. Theory* **43**(6), 1757–1766 (1997)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.