Available online at http://scik.org
J. Math. Comput. Sci. 11 (2021), No. 6, 6786-6810
https://doi.org/10.28919/jmcs/6438
ISSN: 1927-5307

CONSTRUCTION OF MATRIX PRODUCT CODES WHERE DEFINING MATRIX IS GENERATOR MATRIX OF SOME BCH CODE OR RS CODE, USING SAGEMATH FUNCTIONS

POOJA G. RAJANI[1,*], ARUNKUMAR R. PATIL[2]

[1]Department of Mathematics, Smt.Chandibai Himathmal Mansukhani College, Ulhasnagar-3, Mumbai, India

[2]Department of Mathematics, Shri Guru Gobind Singhji Institute of Engineering and Technology, Nanded, India

Abstract. For a given finitely many codes, Matrix Product Code (MPC) can generate a code with better dimension and minimum distance. In this paper, SageMath functions are defined and using these functions, matrix product codes (MPC) are constructed with defining matrix as the generator matrix of some BHC-code or RS-code.

Keywords: cyclic code; BCH code; RS code.

2010 AMS Subject Classification: 94B15.

1. INTRODUCTION

Let $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l$ be linear codes of length n over $\mathbb{F}_q$. Matrix Product Code $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is a code that consist of all matrix products $[c_1, c_2, ..., c_l].\mathcal{A}$, where $c_i \in \mathcal{C}_i$ for $1 \leq i \leq l$ and $\mathcal{A}$ is some $l \times s$ matrix over $\mathbb{F}_q$. Matrix $\mathcal{A}$ is called defining matrix of code $\mathcal{C}$. Various research has been done with Matrix Product Codes. To specify a few, in [1] it is shown that the well known constructions like $(u|u+v), (u+v+w|2u+v|u)$ are Matrix Product Codes. Further it is shown that General Reed- Muller codes are also Matrix Product Codes. In [8] Matrix Product Linear Complementray Dual Codes (MPLCD Codes) are constructed with

*Corresponding author

E-mail address: poojarochani.d@gmail.com

Received June 30, 2021

defining matrix as Quasi orthogonal matrix. In [11] dual containing Matrix Product Codes are constructed. In [3] squares of Matrix Product Codes are calculated with defining matrix as Vandermonde matrix or MS_p matrix with $p-$prime($MS_p = [\binom{p-i}{j-1}$ modulo p]) and it is shown that square of such codes is also a Matrix Product Code of similar kind.

Our contribution in this paper is to find the Matrix Product Codes where defining matrix is the generator matrix of some BCH code or RS code, using SageMath functions. The paper is organized as follows: In section (2) basic concepts of linear codes, cyclic codes, BCH codes, RS codes and Matrix Product Codes are given. In section (3) our aim is to find the SageMath function that accepts the values of n,q and display the list of all $q-$ cyclotomic cosets and their corresponding irreducible factors of $x^n - 1$ over $\mathbb{F}_q$. Once it is done, some other SageMath functions are defined which accepts the list of elements from distinct cyclotomic cosets and display the parameters and type(BCH or non BCH) of associated cyclic code. We have also defined a function which accepts the values of n, q, m and prints all cyclic codes of length n over $\mathbb{F}_q$ of dimension m. In section (4) and (5) we have illustrated the Matrix Product Codes with defining matrix as a generator matrix of some BCH code and RS code respectively.

2. PRELIMINARIES

Let $\mathbb{F}_q$ denotes the finite field with q elements where q is a power of a prime p. $k-$ dimensional subspace $\mathcal{C}$ of $\mathbb{F}_q^n$ is $[n,k]-$ linear code over $\mathbb{F}_q$. Vectors in code $\mathcal{C}$ are called codewords. The Hamming distance between any two codewords x, y is number of non matching coordinates of x, y. Distance of a code $\mathcal{C}$ is minimum of all possible Hamming distances between any two codewords or minimum of weight of codewords, where weight of a codeword is the number of non zero coordinates in a codeword. $k-$ dimensional subspace of $\mathbb{F}_q^n$ with minimum distance d is referred as $[n,k,d]-$ linear code over $\mathbb{F}_q$. Dual of a code $\mathcal{C}$, $\mathcal{C}^\perp$ is a subspace of $\mathbb{F}_q^n$ that contains all those words which are orthogonal to each and every codeword of $\mathcal{C}$.

A linear code $\mathcal{C}$ is cyclic if for any codeword in $\mathcal{C}$, a cyclic shift is also a codeword in $\mathcal{C}$. Now onwards we will assume that $(n,q) = 1$. Now lets recall the concepts of cyclic codes, their defining and generating sets as given in [7], [9] and [2].

We find the smallest $m \in \mathbb{Z}^+$ so that $n|q^m - 1$. As we know that for any divisor l of order of a cyclic group, there exist an element of order l in a group. Here, $(\mathbb{F}_{q^m})^*$ is a cyclic group

of order q^m-1 and $n|q^m-1$ implies that there exist an element of order n in an extension $\mathbb{F}_{q^m}$ of $\mathbb{F}_q$ (say β). This β is primitive n^{th} root of unity. Hence, $(x^n-1)=\prod_{s\in\{1,2,...,n\}}(x-\beta^s)$. It is well known fact that $\beta^s,\beta^{sq},...,\beta^{sq^{m-1}}$ have same minimal polynomial over $\mathbb{F}_q$, $\mathcal{MP}_s(X)=\prod_{i\in\{0,1,...m-1\}}(x-\beta^{sq^i})$. A $q-$ cyclotomic coset modulo n containing s is defined as $C_s=\{sq^k(\text{modulo } n)|k=0,1,...,m-1\}$. These cyclotomic cosets are disjoint and cover $\mathbb{Z}_n$. For every cyclotomic coset, there is an irreducible factor of x^n-1 and vice a versa (i.e $\mathcal{MP}_s(X)\leftrightarrow C_s$). It is also well know fact that every cyclic code over $\mathbb{F}_q$ of length n is associated to an ideal $\langle g(x)\rangle$ of $\frac{\mathbb{F}_q[x]}{\langle x^n-1\rangle}$, where $g(x)$ is a monic irreducible factor of x^n-1 over $\mathbb{F}_q$ called as generator polynomial of a cyclic code. As $g(x)$ is a product of irreducible factors of x^n-1 and each such irreducible factor corresponds to some cyclotomic coset in $\mathbb{Z}_n$ hence, each cyclic code corresponds to the union of cyclotomic cosets in $\mathbb{Z}_n$. Now one can define Defining set of a cyclic code $\mathcal{C}$ with generator polynomial $g(x)$ as $S=\{s\in\mathbb{Z}_n|\beta^s \text{ is a root of } g(x)\}$ and Generating set is defined as $\mathbb{Z}_n-S=T=\{t\in\mathbb{Z}_n|\beta^t \text{ is not a root of } g(x)\}$.

It is a well known fact that if $\mathcal{C}$ is a cyclic code of length n over $\mathbb{F}_q$ and if there exist consecutive $d-1$ integers in defining set S of $\mathcal{C}$, then $d(\mathcal{C})\geq d$. Here d is called a BCH bound. Further if the codeword corresponding to its generator polynomial is itself of weight d then $d(\mathcal{C})=d$.

A BCH code of length n over $\mathbb{F}_q$ is a cyclic code with generator polynomial as $g(x)=lcm(\mathcal{MP}_a(x),\mathcal{MP}_{a+1}(x),....,\mathcal{MP}_{a+d-2}(x))$,for some integer a, where $\mathcal{MP}_i(x)$ is the minimal polynomial of β^i over $\mathbb{F}_q$, for $i=a$ to $a+d-2$. Let $\mathcal{C}$ be a BCH code as above, then $\{a,a+1,...,a+d-2\}\subseteq J$. So, $d-1$ consecutive integers are inside the defining set of $\mathcal{C}$. Hence by BCH bound, the minimum distance of code $\mathcal{C}$ is atleast d. So, $\mathcal{C}$ is $[n,n-deg(g(x)),\geq d]-$ BCH code. Further, if the codeword corresponding to generator polynomial of $\mathcal{C}$ is itself of weight equal to d then, $\mathcal{C}$ is $[n,n-deg(g(x)),d]-$BCH code.

Now take the special case of BCH codes when $m=1$ and $n=q-1$. In this case β is primitive $(q-1)^{th}$ root of unity in $\mathbb{F}_q$. So, $(x^{q-1}-1)=(x-\beta)(x-\beta^2)(x-\beta^3)...(x-\beta^{q-1})$ is the factorization of $(x^{q-1}-1)$ over $\mathbb{F}_q$. In this case the minimal polynomial of β^i over $\mathbb{F}_q$ is $\mathcal{MP}_i(x)=(x-\beta^i)$, for $1\leq i\leq q-1$. Hence, $g(x)=lcm(\mathcal{MP}_a(x),\mathcal{MP}_{a+1}(x),....,\mathcal{MP}_{a+d-2}(x))=$

$(x-\beta^a)(x-\beta^{a+1})...(x-\beta^{a+d-2})$. RS code of length $q-1$ over $\mathbb{F}_q$ is the BCH code whose generator polynomial is $g(x)=(x-\beta^a).(x-\beta^{a+1})....(x-\beta^{a+d-2})$ where β is primitive element of $\mathbb{F}_q$.

Definition 2.1 (Matrix product code)**.** *Let $\mathcal{A}=(a_{ij})$ be a $l\times s$ matrix over $\mathbb{F}_q$ and let $\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_l$ be linear codes of length n over $\mathbb{F}_q$. The Matrix Product Code $\mathcal{C}=[\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_l].\mathcal{A}$ is the set of all ns tuples obtained by writing columnwise entries of $n\times s$ matrix $[c_1,c_2,...,c_l].\mathcal{A}$, where $c_i\in C_i,\forall 1\leq i\leq l$. Matrix $\mathcal{A}$ is called Defining matrix of code $\mathcal{C}$.*

The codewords of $\mathcal{C}=[\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_l].\mathcal{A}$ are *ns* tuples obtained by writing coloumnwise entries of the following matrices.

$$\begin{pmatrix} c_{11}a_{11}+c_{12}a_{21}+...+c_{1l}a_{l1} & ..... & c_{11}a_{1s}+c_{12}a_{2s}+...+c_{1l}a_{ls} \\ \vdots & \vdots & \vdots \\ c_{n1}a_{11}+c_{n2}a_{21}+...+c_{nl}a_{l1} & ..... & c_{n1}a_{1s}+c_{n2}a_{2s}+...+c_{nl}a_{ls} \end{pmatrix}$$

Following proposition in [4] and [5] finds the parameters of Matrix Product Codes

Proposition 2.2. *Let $\mathcal{A}=(a_{ij})$ be a $l\times s$ FRR(Full Row Rank) matrix over $\mathbb{F}_q$. $\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_l$ be linear codes of length n over $\mathbb{F}_q$ with dimensions $k_1,k_2,...,k_l$ and mimimum distances $d_1,d_2,...d_l$ respectively. Let S_i be the subspace of $\mathbb{F}_q^s$ generated by first i rows of matrix $\mathcal{A}$, for each $1\leq i\leq l$. Then $\mathcal{C}=[\mathcal{C}_1,\mathcal{C}_2,...,\mathcal{C}_l].\mathcal{A}$ is $[ns,k_1+k_2+...k_l,\geq min\{d_1d(S_1),d_2d(S_2),...d_ld(S_l)\}]$ Matrix Product Code.*

Further, If $\mathcal{C}_1\supseteq\mathcal{C}_2\supseteq....\supseteq\mathcal{C}_l$ then $d(\mathcal{C})=min\{d_1d(S_1),d_2d(S_2),...d_ld(S_l)\}$

3. SageMath functions

(1) User defined function which accepts the values of n,q and s, returns $q-$ cyclotomic coset modulo n containing s.

```
def cyclotomic_coset(s,q,n):
    c=[s]
    f=0
    for j in range (1,n):
        for i in range (0,j):
```

```
            if ((s*q^j) % n == c[i]):
             f=1
             break
        if (f==1):
            break
        c.append((s*q^j)%n)
    return c
Input:
cyclotomic_coset(3,5,26)

```

Output: $[3, 15, 23, 11]$

(2) User defined function which accepts the values of n and q, returns all $q-$ cyclotomic cosets modulo n

```
def all_cyclotomic_cosets(q,n):
    c=[0]
    show(c)
    for j in range(1,n):
        if (j in c):
            continue
        else:
            show(cyclotomic_coset(j,q,n))
            c.extend(cyclotomic_coset(j,q,n))
Input:
all_cyclotomic_cosets(5,26)

```

Output:

[0]

[1,5,25,21]

[2,10,24,16]

[3,15,23,11]

[4,20,22,6]

[7,9,19,17]

[8,14,18,12]

[13]

(3) User defined function which accepts the values of n and q, prints all $q-$ cyclotomic cosets modulo n with their associated irreducible factors of $x^n - 1$ over $\mathbb{F}_q$.

```
def all_cyclotomic_cosets_irreducible_factors(q,n):
    i=1
    while(i>0):
        if ((q^i-1) % n == 0):
            break
        else:
            i=i+1
    m=i
    F.<a>=FiniteField(q^m)
    S.<x>=F[]
    k=(q^m-1)/n
    b=a^k
    c=[0]
    cc=[[0]]
    p=[(x-b^0)]
    d=cyclotomic_coset(1,q,n)
    cc.append(d)
```

```
18      f1=x^0
19      for i in d:
20          f1=f1*(x-b^i)
21      p.append(f1)
22      c.extend(d)
23      for j in range(2,n):
24          if (j in c):
25              continue
26          else:
27              d=cyclotomic_coset(j,q,n)
28              cc.append(d)
29              f1=x^0
30              for i in d:
31                  f1=f1*(x-b^i)
32              p.append(f1)
33              c.extend(cyclotomic_coset(j,q,n))
34      T=table([cc,p])
35      show(T.transpose())
36
37  Input1:
38  all_cyclotomic_cosets_irreducible_factors(5,26)
39  Input2:
40  all_cyclotomic_cosets_irreducible_factors(4,63)
41  Input3:
42  all_cyclotomic_cosets_irreducible_factors(7,64)
43
```

Output1:

$[0]$	$x+4$
$[1,5,25,21]$	$x^4+3x^3+x^2+3x+1$
$[2,10,24,16]$	x^4+3x^3+3x+1
$[3,15,23,11]$	x^4+2x^3+2x+1
$[4,20,22,6]$	$x^4+x^3+4x^2+x+1$
$[7,9,19,17]$	$x^4+4x^3+4x^2+4x+1$
$[8,14,18,12]$	$x^4+2x^3+x^2+2x+1$
$[13]$	$x+1$

Output2:

$[0]$	$x+1$
$[1,4,16]$	$x^3+x^2+x+a^3+a^2+a$
$[2,8,32]$	$x^3+x^2+x+a^3+a^2+a+1$
$[3,12,48]$	$x^3+\left(a^3+a^2+a\right)x^2+1$
$[5,20,17]$	$x^3+x^2+\left(a^3+a^2+a\right)x+a^3+a^2+a+1$
$[6,24,33]$	$x^3+\left(a^3+a^2+a+1\right)x^2+1$
$[7,28,49]$	$x^3+a^3+a^2+a$
$[9,36,18]$	x^3+x+1
$[10,40,34]$	$x^3+x^2+\left(a^3+a^2+a+1\right)x+a^3+a^2+a$
$[11,44,50]$	$x^3+\left(a^3+a^2+a+1\right)x^2+\left(a^3+a^2+a\right)x+a^3+a^2+a+1$
$[13,52,19]$	$x^3+\left(a^3+a^2+a+1\right)x^2+x+a^3+a^2+a$
$[14,56,35]$	$x^3+a^3+a^2+a+1$
$[15,60,51]$	$x^3+\left(a^3+a^2+a\right)x+1$
$[21]$	$x+a^3+a^2+a$
$[22,25,37]$	$x^3+\left(a^3+a^2+a\right)x^2+\left(a^3+a^2+a+1\right)x+a^3+a^2+a$
$[23,29,53]$	$x^3+\left(a^3+a^2+a\right)x^2+\left(a^3+a^2+a+1\right)x+a^3+a^2+a+1$
$[26,41,38]$	$x^3+\left(a^3+a^2+a\right)x^2+x+a^3+a^2+a+1$
$[27,45,54]$	x^3+x^2+1
$[30,57,39]$	$x^3+\left(a^3+a^2+a+1\right)x+1$
$[31,61,55]$	$x^3+\left(a^3+a^2+a\right)x^2+\left(a^3+a^2+a\right)x+a^3+a^2+a$
$[42]$	$x+a^3+a^2+a+1$
$[43,46,58]$	$x^3+\left(a^3+a^2+a+1\right)x^2+\left(a^3+a^2+a\right)x+a^3+a^2+a$
$[47,62,59]$	$x^3+\left(a^3+a^2+a+1\right)x^2+\left(a^3+a^2+a+1\right)x+a^3+a^2+a+1$

Output3:

$[0]$	$x+6$
$[1,7,49,23,33,39,17,55]$	x^8+x^4+6
$[2,14,34,46]$	x^4+x^2+6
$[3,21,19,5,35,53,51,37]$	x^8+4x^4+6
$[4,28]$	x^2+x+6
$[6,42,38,10]$	x^4+4x^2+6
$[8,56]$	x^2+4x+1
$[9,63,57,15,41,31,25,47]$	x^8+6x^4+6
$[11,13,27,61,43,45,59,29]$	x^8+3x^4+6
$[12,20]$	x^2+4x+6
$[16,48]$	x^2+1
$[18,62,50,30]$	x^4+6x^2+6
$[22,26,54,58]$	x^4+3x^2+6
$[24,40]$	x^2+3x+1
$[32]$	$x+1$
$[36,60]$	x^2+6x+6
$[44,52]$	x^2+3x+6

From now onwards, we will be using list l whose elements are the integers from distinct $q-$ cyclotomic cosets modulo n and the associated cyclic code will be the code whose defining set is union of $q-$ cyclotomic cosets modulo n containing $l[i]$, for all i from 0 to (length of list l -1)

(4) User defined function that accepts the list l, values of n and q, returns the maximum size of consecutive integers in defining set of the associated cyclic code.

```
def max_size_consecutive_integers(l,q,n):
    e=[]
    for i in range(0,len(l)):
        e.extend(cyclotomic_coset(l[i],q,n))
```

```
    e.sort()
    s=1
    c=1
    for i in range(0,len(e)-1):
        if (e[i+1] == (e[i]+1)):
            c=c+1
            if (s<c):
                s=c
        else:
            c=1
    if (n-1 not in e or 0 not in e):
        return s
    else:
        f=0
        for i in range(0,s-1):
            if (e[i+1]!=e[i]+1):
                f=1
        g=0
        for i in range(len(e)-s,len(e)-1):
            if (e[i+1]!=e[i]+1):
                g=1
        if(f==1 and g==1):
            return s
        else:
            s1=1
            i=0
            while(e[i+1]==e[i]+1):
                s1=s1+1
```

```
                i=i+1
            s2=1
            i=len(e)-1
            while(e[i]==e[i-1]+1):
                s2=s2+1
                i=i-1
            return s1+s2
  Input:
  l=[0,1,2,3,4]
  max_size_consecutive_integers(l,5,26)

```

Output: 13

(5) User defined function that accepts list l, values of n and q, returns the generating polynomial of the associated cyclic code.

```
def gen_poly(l,q,n):
    e=[]
    for i in range(0,len(l)):
        e.extend(cyclotomic_coset(l[i],q,n))
    i=1
    while(i>0):
        if ((q^i-1) % n == 0):
            break
        else:
            i=i+1
    m=i
    F.<a>=FiniteField(q^m)
```

```
    S.<x>=F[]
    k=(q^m-1)/n
    b=a^k
    f1=x^0
    for i in e:
         f1=f1*(x-b^i)
    return f1
Input:
l=[0,1,2,3,4]
gen_poly(l,5,26)
```

Output:

$x^{17}+3*x^{16}+2*x^{14}+3*x^{13}+x^{12}+3*x^{11}+3*x^9+2*x^8+2*x^6+4*x^5+2*x^4+3*x^3+2*x+4$

(6) User defined function that accepts list l, values of n and q, returns the minimum distance of the associated cyclic code. (It returns the BCH bound, if the generator polynomial is of weight equal to BCH bound otherwise it returns "atleast" BCH bound)

```
def min_dist(l,q,n):
    a=max_size_consecutive_integers(l,q,n)
    g=gen_poly(l,q,n)
    b=len(g.coefficients())
    if(a+1 == b):
        return a+1
    else:
        return (a+1,"atleast")
Input1:
l=[0,1,2,3,4]
min_dist(l,5,26)
```

```
Input2:
l1=[4,7,8,13]
min_dist(l1,5,26)
```

Output1: 14

Output2: (5, 'atleast')

(7) User defined function that accepts list l, values of n and q, returns the dimension of the associated cyclic code.

```
def dim(l,q,n):
    d=0
    for i in range (0,len(l)):
        d=d+len(cyclotomic_coset(l[i],q,n))
    return n-d
  Input:
  l=[0,1,2,3,4]
  dim(l,5,26)
```

Output: 9

(8) User defined function that accepts list l, values of n and q, returns whether the associated cyclic code is BCH or not.

```
def is_bch(l,q,n):
    ds=[]
    for j in range(0,len(l)):
        ds.extend(cyclotomic_coset(l[j],q,n))
    ds.sort()
    ucc=[]
    f=0
    i=0
    while(i < len(ds)-1):
```

```
10        ucc.extend(cyclotomic_coset(ds[i],q,n))
11        while(ds[i+1]==ds[i]+1):
12            if(ds[i+1] in ucc):
13                i=i+1
14                if(i==len(ds)-1):
15                    break
16            else:
17
    ucc.extend(cyclotomic_coset(ds[i+1],q,n))
18                i=i+1
19                if(i==len(ds)-1):
20                    break
21        ucc.sort()
22        if((ucc[0]==0) and (ds[len(ds)-1]==n-1)):
23            j=len(ds)-1
24            while(ds[j]==ds[j-1]+1):
25                if(ds[j] not in ucc):
26
    ucc.extend(cyclotomic_coset(ds[j],q,n))
27                j=j-1
28            if(ds[j] not in ucc):
29                ucc.extend(cyclotomic_coset(ds[j],q,n))
30            ucc.sort()
31        if(ds==ucc):
32            return(" BCH code")
33            f=1
34            break
35        else:
```

```
            ucc=[]
            i=i+1
    if(f==0 and ds!=[0]):
        return(" non BCH code")
    if(ds==[0]):
        return(" BCH code")
  Input1:
  l=[0,1,2,3,4]
  is_bch(l,5,26)

  Input2:
  l1=[1,2,8]
  is_bch(l1,5,26)
```

Output1: BCH code

Output2: non BCH code

(9) User defined function that accepts list l, values of n and q, returns parameters of the associated cyclic code along with its type(BCH or non BCH) and its generator polynomial

```
def associated_cyclic_code(l,q,n):
    print("The Associated cyclic code is
   [",n,",",dim(l,q,n),",",min_dist(l,q,n),"]-",
   is_bch(l,q,n),"over F",q )
    print("Its generator polynomial is",gen_poly(l,q,n))

  Input1:
  l=[0,1,3,5,7,9,11]
  associated_cyclic_code(l,2,63)
  Input2:
```

```
l=[2,5,8]
associated_cyclic_code(l,3,16)
Input3:
l=[0,3,9]
associated_cyclic_code(l,5,27)
Input4:
l=[0,1,2,3,4,6,8,24,32,36,44]
associated_cyclic_code(l,7,64)
```

Output1:

The Associated cyclic code is [63 , 29 , 14]-BCH code over F2

Its generator polynomial is $x^{34}+x^{33}+x^{30}+x^{27}+x^{26}+x^{19}+x^{14}+x^{12}+x^{11}+x^{10}+x^6+x^4+x+1$

Output2:

The Associated cyclic code is [16 , 9 , (5, 'atleast')]-BCH code over F3

Its generator polynomial is $x^7+2*x^5+2*x^4+2*x^2+2*x+1$

Output3:

The Associated cyclic code is [27 , 18 , 2]-non BCH code over F5

Its generator polynomial is x^9+4

Output4:

The Associated cyclic code is [64 , 28 , (10, 'atleast')]-non BCH code over F7

Its generator polynomial is $x^{36}+3*x^{35}+3*x^{33}+5*x^{32}+4*x^{31}+6*x^{30}+x^{29}+5*x^{28}+2*x^{27}+2*x^{25}+2*x^{23}+5*x^{22}+3*x^{21}+3*x^{19}+6*x^{18}+6*x^{16}+4*x^{15}+4*x^{13}+4*x^{12}+2*x^{11}+5*x^{10}+3*x^9+x^8+6*x^7+4*x^6+4*x^5+3*x^4+6*x^3+4*x^2+4*x+1$

(10) User defined function that accepts values of m,n and q, returns all possible cyclic codes of length n over $\mathbb{F}_q$ of dimension m with all the basic details (if exists), otherwise it returns the message that cyclic code of given dimension does not exist. Hence, this

function helps to find the BCH code of length n over $\mathbb{F}_q$ of dimension m with maximum Hamming distance

```
def cyclic_codes_of_given_dim(m,q,n):
    c=[0]
    r=[]
    l1=[]
    flag=0
    r.append(c[0])
    for j in range(1,n):
        if (j in c):
            continue
        else:
            t=cyclotomic_coset(j,q,n)
            r.append(t[0])
            c.extend(cyclotomic_coset(j,q,n))
    S=Subsets(r, submultiset=true)
    k=S.cardinality()
    for i in range (1,k-1):
        if(len(S[i])==1):
            d=len(cyclotomic_coset(S[i][0],q,n))
        else:
            d=0
            for j in range (0,len(S[i])):
                d=d+len(cyclotomic_coset(S[i][j],q,n))
        if(d==n-m):
            show(S[i])
            ds=[]
            for j in range(0,len(S[i])):
```

```
                ds.extend(cyclotomic_coset(S[i][j],q,n))
            ds.sort()
            print("Defining set=", ds)
            print("Generator
  polynomial",gen_poly(S[i],q,n))
            print("Minimum Distance =",min_dist(S[i],q,n))
            print("Type:",is_bch(S[i],q,n))
            print(".........................")
            flag=1
    if(flag==0):
        print("There is no cyclic code of length",n,"over
  F",q,"of dimension", m)

 Input1:
 all_cyclotomic_cosets_irreducible_factors(5,21)
 cyclic_codes_of_given_dim(7,5,21)
 Input2:
 all_cyclotomic_cosets_irreducible_factors(4,17)
 cyclic_codes_of_given_dim(6,4,17)
 Input3:
 all_cyclotomic_cosets_irreducible_factors(17,15)
 cyclic_codes_of_given_dim(2,17,15)
```

Output1:

$[0]$	$x+4$
$[1,5,4,20,16,17]$	$x^6+4x^5+3x^4+x^3+3x^2+4x+1$
$[2,10,8,19,11,13]$	$x^6+2x^4+2x^3+2x^2+1$
$[3,15,12,18,6,9]$	$x^6+x^5+x^4+x^3+x^2+x+1$
$[7,14]$	x^2+x+1

$[1,2,7]$

Defining set= $[1,2,4,5,7,8,10,11,13,14,16,17,19,20]$

Generator polynomial=$x^{14}+x^7+1$

Minimum Distance = 3

Type: non BCH code

.....

$[1,3,7]$

Defining set=$[1,3,4,5,6,7,9,12,14,15,16,17,18,20]$

Generator polynomial= $x^{14}+x^{13}+4*x^{12}+2*x^{11}+4*x^{10}+2*x^9+4*x^7+2*x^5+$
$4*x^4+2*x^3+4*x^2+x+1$

Minimum Distance = (6, 'atleast')

Type: BCH code

.....

$[2,3,7]$

Defining set= $[2,3,6,7,8,9,10,11,12,13,14,15,18,19]$

Generator polynomial= $x^{14}+2*x^{13}+4*x^{11}+4*x^9+2*x^8+2*x^7+2*x^6+4*x^5+$
$4*x^3+2*x+1$

Minimum Distance = 11

Type: BCH code

.....

Output2:

$[0]$	$x+1$
$[1,4,16,13]$	$x^4+\left(a^7+a^6+a^4+a^2+a+1\right)x^3+x^2+\left(a^7+a^6+a^4+a^2+a+1\right)x+1$
$[2,8,15,9]$	$x^4+\left(a^7+a^6+a^4+a^2+a\right)x^3+x^2+\left(a^7+a^6+a^4+a^2+a\right)x+1$
$[3,12,14,5]$	$x^4+x^3+\left(a^7+a^6+a^4+a^2+a\right)x^2+x+1$
$[6,7,11,10]$	$x^4+x^3+\left(a^7+a^6+a^4+a^2+a+1\right)x^2+x+1$

There is no cyclic code of length 17 over F4 of dimension 6

Output 3:

$[0]$	$x+16$
$[1,2,4,8]$	$x^4+5x^3+15x^2+11x+1$
$[3,6,12,9]$	$x^4+x^3+x^2+x+1$
$[5,10]$	x^2+x+1
$[7,14,13,11]$	$x^4+11x^3+15x^2+5x+1$

Defining set= [0, 1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14]

Generator polynomial=$x^{13}+16*x^{12}+x^{10}+16*x^9+x^7+16*x^6+x^4+16*x^3+x+16$

Minimum Distance = 10

Type: BCH code

4. Matrix Product Code with Defining matrix as generator matrix of some BHC Code

Proposition 4.1. *Let $\mathcal{A}$ be a generator matrix of some BCH code $\mathcal{C}'$ of length s over $\mathbb{F}_q$ with generator polynomial*
$g(x) = lcm(\mathcal{MP}_a(x), \mathcal{MP}_{a+1}(x), ..., \mathcal{MP}_{a+d'-2}(x))$, for some integer a. Let $s-deg(g(X)) = l$ and $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l$ be linear codes of length n over $\mathbb{F}_q$ with dimensions $k_1, k_2, ..., k_l$ and mimimum distances $d_1, d_2, ...d_l$ respectively. Then $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1+k_2+...+k_l, \geq min\{d_1d', d_2d', ...d_ld'\}]$ Matrix Product Code. Further, If $\mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq .... \supseteq \mathcal{C}_l$ and the codeword corresponding to g(x) is itself of weight d' then $[\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1+k_2+...+k_l, min\{d_1d', d_2d', ...d_ld'\}]$ Matrix Product Code.

Proof. $\mathcal{A}$ being the generator matrix, it is of Full Row Rank (FRR). $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l$ are linear codes of length n over $\mathbb{F}_q$ with dimensions $k_1, k_2, ..., k_l$ and mimimum distances $d_1, d_2, ...d_l$ respectively. Let S_i be the subspace of $\mathbb{F}_q^s$ generated by first i rows of matrix $\mathcal{A}$, for each $1 \leq i \leq l$. By proposition 2.2, $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1+k_2+...k_l, \geq min\{d_1d(S_1), d_2d(S_2), ...d_ld(S_l)\}]$ Matrix Product Code. Note that $S_1 \subseteq S_2 \subseteq ... \subseteq S_l$ hence we have

$$d(S_1) \geq d(S_2) \geq ... \geq d(S_l) = d(\mathcal{C}') \tag{1}$$

Now $\mathcal{C}'$ is a BCH code of length s over $\mathbb{F}_q$ with generator polynomial $g(x) = lcm(\mathcal{MP}_a(x), \mathcal{MP}_{a+1}(x), ..., \mathcal{MP}_{a+d'-2}(x))$. Hence, $\mathcal{C}'$ is $[s, s-deg(g(X)), \geq d']-$ BCH code. So from $(1), d(S_i) \geq d'$ for all $1 \leq i \leq l$ which implies that $d_i d(S_i) \geq d_i d'$, for all $1 \leq i \leq l$. Hence we have $min\{d_1 d(S_1), d_2 d(S_2), ...d_l d(S_l)\} \geq min\{d_1 d', d_2 d', ...d_l d'\}$. Hence, $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1 + k_2 + ...k_l, \geq min\{d_1 d', d_2 d', ...d_l d'\}]$ Matrix Product Code.

Further if $\mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq .... \supseteq \mathcal{C}_l$ then by second part of proposition 2.2, $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1 + k_2 + ...k_l, min\{d_1 d(S_1), d_2 d(S_2), ...d_l d(S_l)\}]$ Matrix Product Code. It is also given that the codeword corresponding to g(x) is itself of weight d'. Hence $\mathcal{C}'$ is $[s, s - deg(g(x)), d']-$ BCH code. Also $d(S_1) =$ weight of codeword corresponding to $g(x) = d'$. Hence from (1), we have $d' = d(S_1) \geq d(S_2) \geq ... \geq d(S_l) = d(\mathcal{C}') = d'$

Therefore $d' = d(S_1) = d(S_2) = ... = d(S_l) = d(\mathcal{C}') = d'$

Hence, $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_l].\mathcal{A}$ is $[ns, k_1 + k_2 + ... + k_l, min\{d_1 d', d_2 d', ...d_l d'\}]$ Matrix Product Code. □

Example 4.2. *Using above sagemath functions we have,*

$$\mathcal{C}_1 =< x^{10} + x^8 + x^2 + 1 >$$

$$\mathcal{C}_2 =< x^{12} + x^{11} + x^9 + 2x^8 + x^4 + x^3 + x + 2 >$$

$$\mathcal{C}_3 =< x^{13} + 2x^{11} + x^{10} + x^9 + x^8 + x^5 + 2x^3 + x^2 + x + 1 >$$

are cyclic codes of with parameters [16 , 6 , 4], [16 , 4 , 8] and [16 , 3 , 10] respectively over $\mathbb{F}_3$ *with* $\mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq \mathcal{C}_3$.

Also by using sagemath function for finding all codes with given dimension, we have

$\mathcal{C}' =< x^5 + x^4 + x^3 + 2x^2 + 1 >$ *is [8,3,5] - BCH code with maximum value of minimum distance among all BCH codes of length 8 and dimension 3. Its generator matrix is*

$$\mathcal{A}' = \begin{bmatrix} 1 & 0 & 2 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 2 & 1 & 1 & 1 \end{bmatrix}.$$

$\mathcal{C}'' =< x^4 + 2x^3 + 2x + 2 >$ *is [8,4,4]- BCH code with maximum value of minimum distance among all BCH codes of length 8 and dimension 4. Its generator matrix is*

$\mathcal{A}'' = \begin{bmatrix} 2 & 2 & 0 & 2 & 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 & 2 & 1 & 0 & 0 \\ 0 & 0 & 2 & 2 & 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 2 & 1 \end{bmatrix}$. *Hence, from proposition 4.1, we have the following*

MPC	*Parameter*
$[\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}'$	$[128, 13, 20]$
$[\mathcal{C}_2, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}'$	$[128, 11, 40]$
$[\mathcal{C}_3, \mathcal{C}_3, \mathcal{C}_3]\mathcal{A}'$	$[128, 9, 50]$
$[\mathcal{C}_1, \mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}''$	$[128, 19, 16]$
$[\mathcal{C}_2, \mathcal{C}_2, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}''$	$[128, 15, 32]$
$[\mathcal{C}_3, \mathcal{C}_3, \mathcal{C}_3, \mathcal{C}_3]\mathcal{A}''$	$[128, 12, 40]$

Example 4.3. *Using above sagemath functions we have,*

$$\mathcal{C}_1 = < x^5 + (a^2+a)x^4 + x^2 + a^2 + a + 1 >$$

$$\mathcal{C}_2 = < x^7 + (a^2+a+1)x^6 + x^5 + x^3 + (a^2+a+1)x + a^2 + a >$$

$$\mathcal{C}_3 = < x^8 + (a^2+a+1)x^6 + (a^2+a+1)x^5 + x^4 + (a^2+a+1)x^3 + (a^2+a+1) * x^2 + 1 >$$

are cyclic codes with parameters [15 , 10 , 4], [15 , 8 , 6] and [15 , 7 , 7] respectively over $\mathbb{F}_4$ *with* $\mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq \mathcal{C}_3$.

Also by using sagemath function for finding all codes with given dimension, we have

$\mathcal{C}' = < x^4 + x^3 + x^2 + 1 >$ *is [7,3,4] - BCH code with maximum value of minimum distance among all BCH codes of length 7 and dimension 3. Its generator matrix is* $\mathcal{A}' = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$.

$\mathcal{C}'' = < x^2 + (a^2+a) * x + 1 >$ *is [5,3,3]- BCH code with maximum value of minimum distance among all BCH codes of length 5 and dimension 3. Its generator matrix is*

$\mathcal{A}'' = \begin{bmatrix} 1 & a^2+a & 1 & 0 & 0 \\ 0 & 1 & a^2+a & 1 & 0 \\ 0 & 0 & 1 & a^2+a & 1 \end{bmatrix}$. *Hence, from proposition 4.1, we have the following*

MPC	*Parameter*
$[\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}'$	$[105, 25, 16]$
$[\mathcal{C}_2, \mathcal{C}_2, \mathcal{C}_3]\mathcal{A}'$	$[105, 23, 24]$
$[\mathcal{C}_3, \mathcal{C}_3, \mathcal{C}_3]\mathcal{A}'$	$[105, 21, 28]$
$[\mathcal{C}_1, \mathcal{C}_1, \mathcal{C}_2]\mathcal{A}''$	$[75, 28, 12]$
$[\mathcal{C}_2, \mathcal{C}_2, \mathcal{C}_2]\mathcal{A}''$	$[75, 24, 18]$
$[\mathcal{C}_3, \mathcal{C}_3, \mathcal{C}_3]\mathcal{A}''$	$[75, 21, 21]$

5. Matrix Product Codes with Defining Matrix as Generator Matrix of RS Codes

Recall that RS code of length $q-1$ over $\mathbb{F}_q$ is the BCH code whose generator polynomial is $g(x) = (x-\beta^a).(x-\beta^{a+1})....(x-\beta^{a+d-2})$ where β is primitive element of $\mathbb{F}_q$. Above RS code is $[q-1, q-d, d]-$ linear code over $\mathbb{F}_q$. RS codes are MDS codes.

Proposition 5.1. *Let $\mathcal{C}'$ be the RS code of length $q-1$ over $\mathbb{F}_q$ with generator polynomial $g(x) = (x-\beta^a)(x-\beta^{a+1})(x-\beta^{a+2})...(x-\beta^{a+d'-2})$, where β is primitive element of $\mathbb{F}_q$ then the codeword in $\mathcal{C}'$ corresponding to generator polynomial $g(x)$ is of weight d'.*

Proof. As $\mathcal{C}'$ is RS-code of length $q-1$ over $\mathbb{F}_q$, with generator polynomial $g(x) = (x-\beta^a)(x-\beta^{a+1})(x-\beta^{a+2})...(x-\beta^{a+d'-2})$.
$\mathcal{C}'$ is $[q-1, q-d', d']-$ linear code over $\mathbb{F}_q$. $g(x)$ being a degree $d'-1$ polynomial, it has d' coefficients. If any of the coefficient of $g(x)$ is zero then the corresponding codeword to $g(x)$ in $\mathcal{C}'$ is of weight $< d'$, which is contradiction to the fact that $d(\mathcal{C}') = d'$. Hence, the codeword corresponding to $g(x)$ in $\mathcal{C}'$ is of weight d' □

With this observation, we have the following corollary of proposition 4.1 as below

corollary 5.2. *$\mathcal{A}$ be a generator matrix of some RS code $\mathcal{C}'$ of length $q-1$ over $\mathbb{F}_q$ with generator polynomial $g(x) = (x-\beta^a)(x-\beta^{a+1})(x-\beta^{a+2})...(x-\beta^{a+d'-2})$. Let $\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{q-d'}$ be $q-d'$ linear codes of length n over $\mathbb{F}_q$ with dimensions $k_1, k_2, ..., k_{q-d'}$ and minimum distances $d_1, d_2, ..., d_{q-d'}$ respectively. Then, $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{q-d'}].\mathcal{A}$ is $[n(q-1), k_1+k_2+ ... + k_{q-d'}, \geq min\{d_1d', d_2d', ..., d_{q-d'}d'\}]$ Matrix Product Code. Further, if $\mathcal{C}_1 \supseteq \mathcal{C}_2 \supseteq ... \supseteq \mathcal{C}_{q-d'}$ then as codeword corresponding to $g(x)$ is of weight d' (by previous proposition), $\mathcal{C} = [\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_{q-d'}].\mathcal{A}$ is $[n(q-1), k_1+k_2+...+k_{q-d'}, min\{d_1d', d_2d', ..., d_{q-d'}d'\}]$ Matrix Product Code.*

Example 5.3. *Let β be the primitive element of $\mathbb{F}_{4^2}$. Following are various RS codes of length $15 = 4^2-1$ over $\mathbb{F}_{4^2}$ with all possible distances*

RS codes	*Parameters*
$\mathcal{C}_1 = < \prod_{i=1}^{1}(X-\beta^i) >$	$[15,14,2]$
$\mathcal{C}_2 = < \prod_{i=1}^{2}(X-\beta^i) >$	$[15,13,3]$
$\mathcal{C}_3 = < \prod_{i=1}^{3}(X-\beta^i) >$	$[15,12,4]$
$\vdots$	$\vdots$
$\mathcal{C}_{14} = < \prod_{i=1}^{14}(X-\beta^i) >$	$[15,1,15]$

Now, let $\mathcal{A}$ be the generator matrix of RS code $\mathcal{C}' = \mathcal{C}_8 = < \prod_{i=1}^{8}(X-\beta^i) >$ then $\mathcal{C}'$ is $[15,7,9]-$RS code.

Let $C_1 = \mathcal{C}_8, C_2 = \mathcal{C}_9, C_3 = \mathcal{C}_{10}, C_4 = \mathcal{C}_{11}, C_5 = \mathcal{C}_{12}, C_6 = \mathcal{C}_{13}$ and $C_7 = \mathcal{C}_{14}$, then by corollary 5.2, $\mathcal{C} = [C_1, C_2, ..., C_7].\mathcal{A}$ is $[225,28,81]-$Matrix Product Code.

If $\mathcal{C}' = C_1 = C_2 = ... = C_7 = \mathcal{C}_8$ then $\mathcal{C} = [C_1, C_2, ..., C_7].\mathcal{A}$ is $[225,49,81]-$Matrix Product Code.

If $\mathcal{C}' = C_1 = C_2 = ... = C_6 = \mathcal{C}_9$ then $\mathcal{C} = [C_1, C_2, ..., C_6].\mathcal{A}$ is $[225,36,100]-$Matrix Product Code.

In general, If $\mathcal{C}' = C_1 = C_2 = ... = C_{q-d'} = \mathcal{C}_{d'-1}$ *then* $\mathcal{C} = [C_1, C_2, ..., C_{q-d'}]$.$\mathcal{A}$ *is* $[(q-1)^2, (q-d')^2, d'^2]-$ *Matrix Product Code.*

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] T. Blackmore and G. H. Norton, Matrix-Product Codes over Fq. Appl. Algebra Eng. Commun. Comput. 12(6) (2001), 477-500.

[2] J. I. Cascudo, On squares of cyclic codes. IEEE Trans. Inform. Theory, 65(2) (2019), 1034-1047.

[3] I. Cascudo, J. S. Gundersen, D. Ruano, Squares of matrix-product codes. Finite Fields Appl. 62 (2020), 101606.

[4] H. Fernando, L. Kristine, R. Diego, Construction and decoding of matrix-product codes from nested codes. Applicable Algebra in Engineering, Commun. Comput. 20(5) (2009), 467.

[5] O. Ferruh, S. Henning, Note on Niederreiter-Xing's Propagation Rule for Linear Codes. Applicable Algebra in Engineering, Commun. Comput. 13(1) (2002), 53-56.

[6] W. J. Thomas, Abstract algebra: theory and applications, 2020.

[7] S. Ling, C. Xing, Coding Theory: A First Course. Cambridge University Press, Austin State University Press, 2004.

[8] X. Liu, H. Liu, Matrix-product complementary dual codes. arXiv preprint arXiv:1604.03774, 2016.

[9] F. J. MacWilliams, N. J. A. Sloane, The Theory of Error-Correcting Codes. North-Holland, New York, 1978.

[10] S. G. Lee, et al. Linear Algebra, BigBook, 2018.

[11] H. Song, L. B. Guo, L. D. Lv, G. Chen, New Quantum Codes From Matrix-Product Codes, Procedia Computer Sci. 154 (2019), 686–692.