IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

"Computations in Macaulay2 to construct an algebraic system associated to a given cyclic code and a positive integer $'w'$, whose solutions are in bijection with codewords of weight less than or equal to $'w'$."

Dr. Arunkumar Patil[1]
Department of Mathematics, S.G.G.S. Institute of Engineering and Technology Nanded, India

Pooja Rajani[2]
Department of Mathematics, Smt Chandibai Himathmal Mansukhani College Ulhasnagar, India

ABSTRACT

This paper aims at constructing a user defined functions in Macaulay2 which accepts values of $'n'$ and $'q'$ with $\gcd(n, q) = 1$ and displays all q − cyclotomic cosets modulo n. This list of cyclotomic cosets can help in describinga cyclic code in Macaulay2 by selecting a representative from each cyclotomic coset in its defining set. For a given cyclic code $'C'$ and a positive integer $'w'$, there is a well-known algebraic system constructed from Newton's identities which are satisfied by elementary symmetric functions of locators of a codeword of weight $'w'$ and coefficients of its Mattson-Solomon polynomial. This paper aims at constructing a user defined function in Macaulay2, which accepts a cyclic code (in terms of list of elements from distinct cyclotomic cosets in its defining set) and a positive integer $'w'$ and it returns the algebraic system described above. Further, the simplified form of this system is also constructed. In a special case when integer w is equal to BCH bound of C, the simplified system is used for computing number of codewords of minimum weight in C, using Gröbner basis.

Keywords: linear codes, Cyclic code, BCH code, Mattson-Solomon polynomial, locators of a codeword.

(I) INTRODUCTION

A cyclic code can be constructed in Macaulay2 by providing a field, generator polynomialand length as parameters to the user defined function 'cyclicCode'. In this paper, we have listed all q − cyclotomic cosets modulo n by constructing user defined functions and then described a cycliccode by selecting a set of cyclotomic cosets in its defining set. For a given cyclic code C and $w \in \mathbb{Z}^+$, there is an associated algebraic system $\eta_C(w)$ of several variables (see [5]) whose solutions are in bijection with codewords in C of weight less or equal $'w'$ and when $'w'$ is the BCH bound of C then $\eta_C(w)$ helps in counting number of codewords of minimum weight. By looking at the importance of thesystem $\eta_C(w)$, we have constructed the user defined function in Macaulay2 whose input is a cyclic code $'C'$ & $w \in \mathbb{Z}^+$,and output is the system $\eta_C(w)$. Its simplified form is also constructed in Macaulay2, whose equations defines an ideal in a polynomial ring of multivariable and using Gröbner basis of this ideal, number of codewords of minimum weight can be computed.

This paper is divided into two main sections. Section (1) is preliminary and section (2) is Computations in Macaulay2. Now, section (1) is divided into three subsections. In subsection (A), necessary concepts required for a linear code, cyclic code and BCH code are revised from [1] and [2]. In subsection (B), basic concepts of Gröbner basis of an ideal are discussed from [3] and in subsection (C), an algebraic system $\eta_C(w)$ associated to a cyclic code C and a positive integer $'w'$is recalled from [4] and [5]. Section (2) shows various Macaulay2 computations to achieve aim of this paper.

(II) Preliminary

A. **Linear codes, cyclic codes and BCH bound.**

We recall the notations, definitions and results mentioned in [1] and [2] about finite fields, linear codes, cyclic codes and BCH bound.

Let $\mathbb{F}_q$ be a finite field with q elementswhere q is prime power. $[n, k]$-linear code C is a subspace of$\mathbb{F}_q^n$ with $\dim C = k$. eectors in $\mathbb{F}_q^n$ are called words and vectors in C are called codewords. Hamming distance between two words p, q is denoted by $d(p,q)$ and it is defined as number of non-matching coordinates of p and q. Minimum distance of a code C is denoted by $d(C)$ and it isdefinedas $d(C) = \min\{d(c', c''): c', c'' \in C\}$. Weight of a word p, $wt(p)$ is defined as $wt(p) = |\{i: p_i \neq 0\}|$. Linearity of a code C implies that $d(C) = \min\{wt(c'): c' \in C\}$. Dual of a code C is defined as $C^{\perp} := \{x \in \mathbb{F}_q^n : < x, c' >= 0, \forall c' \in C\}$.

From now, we will assume that n and q are relatively prime. Let $t \in \mathbb{Z}^+$ be the smallest suchthat $n | q^t - 1$. Now, $(\mathbb{F}_{q^t}^*)$ being a cyclic group of size $q^t - 1$ and $n | q^t - 1$, there is an element of order $'n'$ in an extension $\mathbb{F}_{q^t}$ of $\mathbb{F}_q$ say $'\xi'$. Hence, $x^n - 1 = \prod_{i=1}^{n}(x - \xi^i)$. Now $\xi^i, \xi^{iq}, \xi^{iq^2}, ..., \xi^{iq^{t-1}}$ have same minimal polynomial over $\mathbb{F}_q$, $M_i(x) = \prod_{l=0}^{t-1}(x - \xi^{iq^l})$ which are irreducible factors of $x^n - 1$ and these are in bijection with $q-$cyclotomic cosets modulo n containing $i, C_i = \{i.q^l (mod\ n): 0 \leq l \leq t-1\} \subseteq \mathbb{Z}_n$. These cyclotomic cosets are disjoint and cover entire $\mathbb{Z}_n$.

A code C is cyclic if for any codeword in C, a cyclic shift is also a codeword in C. There is one-to-one correspondence between cyclic codes of length $n$over $\mathbb{F}_q$ and ideals $< g(x) >$ of $\frac{\mathbb{F}_q[x]}{<x^n-1>}$, where $g(x)$ is called a generator polynomial of a cyclic code C, which is a product of monic irreducible factors of $x^n - 1$. As mentioned before that every such irreducible factor of $x^n - 1$ corresponds to a cyclotomic coset hence, every cyclic code is associated to disjoint union of these cyclotomic cosets called a defining set of cyclic code C, that is, $I(C) = \{k \in \mathbb{Z}_n : g(\xi^k) = 0\} \subseteq \mathbb{Z}_n$. The complement of defining set in $\mathbb{Z}_n$ is called a generating set of a cyclic code C, $that\ is$, $J = \{k \in \mathbb{Z}_n : g(\xi^k) \neq 0\}$. If for a cyclic code C, there exist say $'d-1'$ consecutive integers in its defining set then $d(C) \geq d$ and this $'d'$ is called its BCH bound of C.

B. **Gröbner basis of an ideal in a polynomial ring of n variables.**

Now let's overview the basics of Gröbner basis as mentioned in [3].

Let L be a field and $R = L[X] = L[x_1, x_2, ... x_n] = \{f(X) = \sum_{\alpha \in \mathbb{Z}_{\geq 0}^n} a_\alpha X^\alpha : a_\alpha \in L\}$ is called a ring of polynomials over L. Let $\alpha = (\alpha_1, \alpha_2, ... \alpha_n)$ then $X^\alpha = x_1^{\alpha_1} x_2^{\alpha_2} ... x_n^{\alpha_n}$ is called monomial. Monomial order on R is a relation $'>'$ on monomials or on $\mathbb{Z}_{\geq 0}^n$ satisfying (i) $'>'$ is total order (ii) If $\alpha > \beta$ and $\gamma \in \mathbb{Z}_{\geq 0}^n$ then $\alpha + \gamma > \beta + \gamma$. (iii) $'>'$ is well ordering. For example, for $\alpha, \beta \in \mathbb{Z}_{\geq 0}^n$, $\alpha > \beta \Leftrightarrow$ leftmost non zero entry of $\alpha - \beta$ is positive. This order gives importance to higher powers of leftmost variable, called a lex order which is a monomial order. With any monomial ordering, any two terms of a polynomial in R are comparable, so one can talk about term that leads, in a polynomial $f(X) \in L[X]$, denoted by $LT(f)$.

Now, $I \subseteq R$ is an ideal, if (i) additive identity of R is in I (ii) I is closed under addition(iii) For $f \in I$ and $h \in R, hf \in I$. Ideal generated by $f_1, f_2, ..., f_n \in R$ is defined as $< f_1, f_2, ... f_l >= \{\sum^k h_i f_i : h_1, ..., h_k \in R\}$. An ideal $I \subseteq L[x_1, x_2, ..., x_n]$ is called a Monomial ideal if $\exists\ A \subseteq \mathbb{Z}_{\geq 0}^n$ (may be infinite) such that $I = \{\sum_{\alpha \in A} h_\alpha x^\alpha | \ h_\alpha \in L[x_1, x_2, ..., x_n]\}$. We write, $I =< x^\alpha : \alpha \in A >$. By Dickson's lemma, any monomial ideal of ring $L[x_1, x_2, ... x_n]$ is finitely generated. Further, if no monomial in generating set is a multiple of another monomial, then it is called "minimal basis". Now, for a given ideal of ring $L[x_1, x_2, ... x_n]$, initial ideal $in(I)$ is defined as an ideal generated by leading terms of each and every polynomial in an ideal I, $in(I) =< LT(f) : f \in I >$. This being a monomial ideal, has finite generating set, by Dickson's lemma. Hilbert theorem states that, polynomials in an ideal I corresponding to generating monomials of initial ideal of I, $in(I)$, generates an ideal I itself. Hence, every ideal of a polynomial ring $R = L[x_1, x_2, ... x_n]$ is finitely generated.

Now, for a given ideal I and a fix monomial ideal, a finite subset $\mathcal{G} \subseteq I$ is called a Gröbner basis of ideal I if $in(I) =< LT(g) : g \in \mathcal{G} >$. If each term of $g \in \mathcal{G}$ is not a multiple of $LT(g')$, $\forall g' \in \mathcal{G} \backslash \{g\}$, then $\mathcal{G}$ is called a reduced Gröbner basis, which is always unique.

Further, Gröbner basis is used to solve a system of polynomial equations in several variables. In order to solve a system of multivariable polynomial equations, one need to consider the ideal generated by given polynomial equations of the system, in a multivariable polynomial ring. Then, Gröbner basis is to be computed which eliminate variables in a very systematic way (like that of Gaussian elimination method in case of linear equations system). Proven result says, if minimal Gröbner basis reduces to $\{1\}$ then the given system has a solution.

C. **Algebraic system $S_{\mathcal{C}}(w)$, associated to a cyclic code $\mathcal{C}$ and a positive integer $'w'$**

Now lets recall few definitions, propositions and theorems mentioned in [4],[5] and [6] related to an algebraic system, associated to a cyclic code C and a positive integer $'i'$.

Let $\alpha \in \mathbb{F}_q^t$ be such that $O(\alpha) = n$ (under multiplication) and $c = (c_0, c_1, \ldots, c_{n-1}) \in \mathbb{F}_q^n$ is of weight $'w'$. Set of locators of $c = \{Z_1, Z_2, \ldots, Z_i\} = \{\alpha^j : for\ j\ such\ that\ c_j \neq 0\}$. The Mattson-Solomon polynomial of c is given by $A = \sum_{i=1}^{n} A_i Z^{n-i} \in \mathbb{F}_q^t[Z]$ where $A_i = c(\alpha^i), 1 \leq i \leq n$. The elementary symmetric functions of c are denoted by $\sigma_1, \sigma_2, \ldots, \sigma_w$, where $\sigma_k = (-1)^k \sum_{1 \leq j_1 < j_2 < \cdots < j_k \leq w} Z_{j_1} Z_{j_2} \ldots Z_{j_k}, \forall k \in \{1,2,\ldots,w\}$

Now for a cyclic code C and a positive integer $'w'$, the associated algebraic system is given as follows (see [5])

$$\eta_C(w) = \begin{cases} A_{w+1} + A_w\sigma_1 + \cdots + A_1\sigma_w = 0 \\ A_{w+2} + A_{w+1}\sigma_1 + \cdots + A_2\sigma_w = 0 \\ \vdots \\ A_{n+w} + A_{n+w-1}\sigma_1 + \cdots + A_n\sigma_w = 0 \\ A_{iq \bmod n} = A_i, \forall i \in [0, n-1] \\ A_{i+n} = A_i, \forall i \in [0, n-1] \\ A_i = 0, \forall i \in I(\mathcal{C}) \end{cases}$$

In the above system, $A_k's$ and $\sigma_k's$ are variables. If $c \in \mathbb{F}_q^n$ with weight $'w'$, Mattson Solomon polynomial coefficients $A_0, A_1, \ldots A_{n-1}$ and elementary symmetric functions $\sigma_1, \ldots, \sigma_w$, then by proposition1.1 and Theorem 2.1 of [5], above system $\eta_C(w)$ has a solution.

Conversely, by Theorem 2.3 of [5], $(A_0, A_1, \ldots, A_{n-1})$ that are solutions to $\eta_C(w)$ are the Mattson Solomon polynomial coefficients of the codewords of C of weight less than or equal to $'w'$, note that $(A_0, A_1, \ldots, A_{n-1})$ is a solution of the system $\eta_C(w)$, if $\exists\ (\sigma_1, \sigma_2, \ldots, \sigma_w)$ such that $(A_0, A_1, \ldots, A_{n-1}, \sigma_1, \sigma_2, \ldots, \sigma_w)$ is a solution to $\eta_C(w)$.

Let C is a cyclic code of length n with BCH bound say $'w'$. If there are solutions to $\eta_C(w)$, then by above discussion, these solutions are coefficients of Mattson Solomon polynomial of codewords of weight less than or equal to w, but there is no codeword in C of weight strictly less than $'w'$, $'w'$ being a BCH bound. Hence, the minimum distance of C is exactly equal to $'w'$. Hence, system $\eta_C(w)$ helps to determine existence of minimum weight codeword. It is also discussed in [5] that number of minimum weights codewords in C is equal to number of solutions to $\eta_C(w)$.

(III) Computations in Macaulay2

It will be always easier to constructa cyclic code of length n over a finite field $\mathbb{F}_q$ with $(n, q) = 1$, if one has the list of all q cyclotomic coset modulo n. Then, one can choose a set of cyclotomic cosets as per the requirement as the defining set of the required cyclic code. To achieve this, following user defined functionsare constructed in Macaulay2

i) **cyclotomic_coset**: It accepts values of q, n, s and returns q-cyclotomic coset modulo n containing $'s'$

```
i1 : cyclotomic_coset=(s,q,n) ->

     (  c={s} ;
        flag =0 ;
        for j from 1 to (n-1) do
        (
           for i from 0 to (j-1) do
           (if ((s*(q^j)) % n) == c_i then (flag=1;break;));
           if flag==1 then break;
           c=join(c,{((s*(q^j))  % n)});
         );
         c
      )

o1 = FunctionClosure[stdio:1:17-13:3]

o1 : FunctionClosure
```

```
i2 : cyclotomic_coset(3,2,7)

o2 = {3, 6, 5}

o2 : List

i3 : cyclotomic_coset(3,4,55)

o3 = {3, 12, 48, 27, 53, 47, 23, 37, 38, 42}

o3 : List

i4 : cyclotomic_coset(23,7,64)

o4 = {23, 33, 39, 17, 55, 1, 7, 49}

o4 : List
```

ii) **lcc**: It accepts the values of q, n and returns list of all q-cyclotomic coset modulo n

```
i5 : lcc = (q,n) ->

     ( c1={0};
       cc={{0}};
       d=cyclotomic_coset(1,q,n);
       cc=join(cc,{d});
       for i in d do( c1=join(c1,{i}));
       for j from 2 to (n-1) do
       (
          if isMember(j,c1)==true then continue
          else(
                d=cyclotomic_coset(j,q,n);
                cc=join(cc,{d});
                c1=join(c1,d);
              )
       );
       cc
     )

o5 = lcc

o5 : FunctionClosure
```

```
i6 : lcc(2,7)

o6 = {{0}, {1, 2, 4}, {3, 6, 5}}

o6 : List

i7 : lcc(4,55)

o7 = {{0}, {1, 4, 16, 9, 36, 34, 26, 49, 31, 14}, {2, 8, 32, 18, 17, 13, 52, 43, 7, 28}, {3, 12, 48, 27, 53, 47, 23, 37, 38, 42},
{5, 20, 25, 45, 15}, {6, 24, 41, 54, 51, 39, 46, 19, 21, 29}, {10, 40, 50, 35, 30}, {11, 44}, {22, 33}}

o7 : List

i8 : lcc(7,64)

o8 = {{0}, {1, 7, 49, 23, 33, 39, 17, 55}, {2, 14, 34, 46}, {3, 21, 19, 5, 35, 53, 51, 37}, {4, 28}, {6, 42, 38, 10}, {8, 56},
{9, 63, 57, 15, 41, 31, 25, 47}, {11, 13, 27, 61, 43, 45, 59, 29}, {12, 20}, {16, 48}, {18, 62, 50, 30}, {22, 26, 54, 58}, {24,
40}, {32}, {36, 60}, {44, 52}}

o8 : List
```

iii) **defining_set**: It accepts values of l, q, n and return the defining set of a cyclic code of length n over $\mathbb{F}_q$, which is a disjoint union of cyclotomic cosets containing list $'l'$ elements.

```
i9 : defining_set = (l,q,n) ->

     (
       ds={};
       for j from 0 to (length l -1) do
       ( if isMember(l_j,ds)==true then continue else ds=join(ds,cyclotomic_coset(l_j,q,n)));
       ds
     )

o9 = FunctionClosure[stdio:48:15-55:1]

o9 : FunctionClosure
```

```
i10 : defining_set({1},2,7)

o10 = {1, 2, 4}

o10 : List

i11 : defining_set({2,36,44},4,55)

o11 = {2, 8, 32, 18, 17, 13, 52, 43, 7, 28, 36, 34, 26, 49, 31, 14, 1, 4, 16, 9, 44, 11}

o11 : List

i12 : defining_set({2,12,16},7,64)

o12 = {2, 14, 34, 46, 12, 20, 16, 48}

o12 : List
```

Now, let's see one demonstration for constructing a system in Macaulay2, say $\eta_C(3)$, where C is a cyclic code of length 7 over $\mathbb{F}_2$ with defining set $cl(1) = \{1,2,4\}$

$$\eta_C(3) = \begin{cases} A_4 + A_3\sigma_1 + A_2\sigma_2 + A_1\sigma_3 = 0 \\ A_5 + A_4\sigma_1 + A_3\sigma_2 + A_2\sigma_3 = 0 \\ A_6 + A_5\sigma_1 + A_4\sigma_2 + A_3\sigma_3 = 0 \\ A_7 + A_6\sigma_1 + A_5\sigma_2 + A_4\sigma_3 = 0 \\ A_8 + A_7\sigma_1 + A_6\sigma_2 + A_5\sigma_3 = 0 \\ A_9 + A_8\sigma_1 + A_7\sigma_2 + A_6\sigma_3 = 0 \\ A_{10} + A_9\sigma_1 + A_8\sigma_2 + A_7\sigma_3 = 0 \\ A^2 = A \ , A^2 = A \ , A^4 = A \\ A_7 = A_0, A_8 = A_1, A_9 = A_2, A_{10} = A_3 \\ A_1 = A_2 = A_4 = 0 \end{cases}$$

Step1: Construction of expressions of the type $A_{i+w} + \sigma_1 A_{i+w-1} + \cdots + \sigma_w A_i = 0$ (note that, $B's$ are used in place for $\sigma's$)

```
i13 : R=GF(2)[A_0,A_1..A_10,B_1..B_6,MonomialOrder=>Lex]

o13 = R

o13 : PolynomialRing

i14 :
      g={};

i16 : for j from 4 to 10 do
      (
        l1={};
        for i from 1 to 4 do
        (
          l1=join(l1,{A_((j+1)-i)})
        );
        l2={1};
        for i from 1 to 3 do
        (
          l2=join(l2,{B_i})
        );
        f1=0;
        for i from 0 to 3 do
        (
          f1=f1+(l1_i * l2_i)
        );
        g=join(g,{f1});
      )

i17 : g

o17 = {A1B3 + A2B2 + A3B1 + A4, A2B3 + A3B2 + A4B1 + A5, A3B3 + A4B2 + A5B1 + A6, A4B3 + A5
B2 + A6B1 + A7, A5B3 + A6B2 + A7B1 + A8, A6B3 + A7B2 + A8B1 + A9, A7B3 + A8B2 + A9B1 + A10,
```

Step 2: Construction of expressions of the type $A_{iq \bmod n} = A \ , \forall i \in [0, -1]^{q}$

```
i18 : p1 = (l,q,n) ->

        ( l'= lcc(q,n);
          z={};
          for m from 0 to (length(l')-1) do
         (
               h=l'_m_0;
               if (isMember(h,defining_set(l,q,n))) then continue else z=join(z,{h})
         );
          g1={};
          for m in z do
          ( i=1;
            while isMember(((m*(q^i))%n),cyclotomic_coset(m,q,n))==true do
            ( g1=join(g1,{A_((m*(q^i))%n)-(A_m)^(q^i)}); i=i+1; if ((m*(q^i))%n)==m then break));
            g1
        )

o18 = p1

o18 : FunctionClosure

i19 : p1({1},2,7)

o19 = {A_0^2 + A_0, A_3^2 + A_6, A_3^4 + A_5}

o19 : List
```

Step 3: Construction of expressions of the type: $A_{i+n} = A_i, \forall i \in [0, n-1]$

```
i20 : p2 = (n,w) ->

        (
          g2={};
          for k from n to n+w do(g2=join(g2,{A_k-A_(k%n)}));
          g2
        )

o20 = p2

o20 : FunctionClosure

i21 : p2(7,3)

o21 = {A_0 + A_7, A_1 + A_8, A_2 + A_9, A_3 + A_10}

o21 : List
```

Step 4: Construction of expressions of the type: $A_i = 0, \forall i \in I(\mathcal{C})$

```
i22 : P3 = (l) ->

        ( g3={};
          for i in l do(g3=join(g3,{A_i}));
          g3
        )

o22 = P3

o22 : FunctionClosure

i24 : p3 = (l) ->

        ( g3={};
          for i in l do(g3=join(g3,{A_i}));
          g3
        )

o24 = p3

o24 : FunctionClosure

i25 : p3(defining_set({1},2,7))

o25 = {A_1, A_2, A_4}

o25 : List
```

Now, for compilation of all above steps, following user defined function is constructed which accepts values of l, q, n, w and display the algebraic system $\eta_C(w)$ associatedto a cyclic code C of length n over $\mathbb{F}_q$ with defining set as a union of cyclotomic cosets containing elements of list $'l'$

```
i26 : system1 = (l,q,n,w) ->

      ( p3 = (l) ->
        ( g3={};
          for i in l do(g3=join(g3,{A_i}));
          g3
        );
        p2 = (n,w) ->
        ( g2={};
          for k from n to n+w do(g2=join(g2,{A_k-A_(k%n)}));
          g2
        );
        p1 = (l,q,n) ->
        ( g1={};
          for m in l do
          ( i=1;
            while isMember(((m*(q^i))%n),cyclotomic_coset(m,q,n))==true do
            ( g1=join(g1,{A_((m*(q^i))%n)-(A_m)^(q^i)}); i=i+1; if ((m*(q^i))%n)==m then break));
            g1
        );
        a={};
        g={};
        for j from (w+1) to (w+n) do
        (
          l1={};
          for i from 1 to (w+1) do
          (
            l1=join(l1,{A_((j+1)-i)})
          );
          l2={1};
          for i from 1 to w do
          (
            l2=join(l2,{B_i})
          );
          f1=0;
          for i from 0 to w do
          (
            f1=f1+(l1_i * l2_i)
          );
          g=join(g,{f1})
        );
        a=join(a,g);
        l'= lcc(q,n);
        k'={};
        for m from 0 to (length(l')-1) do
        (
               h=l'_m_0;
               if (isMember(h,defining_set(l,q,n))) then continue else k'=join(k',{h})
        );
        a=join(a,p1(k',q,n));
        a=join(a,p2(n,w));
        a=join(a,p3(defining_set(l,q,n)));
        a
      )

o26 = system1

o26 : FunctionClosure
```

```
i27 : system1({1},2,7,3)

o27 = {A_1B_3 + A_2B_2 + A_3B_1 + A_4, A_2B_3 + A_3B_2 + A_4B_1 + A_5, A_3B_3 + A_4B_2 + A_5B_1 + A_6, A_4B_3 + A_5B_2 + A_6B_1 + A_7,
A_5B_3 + A_6B_2 + A_7B_1 + A_8, A_6B_3 + A_7B_2 + A_8B_1 + A_9, A_7B_3 + A_8B_2 + A_9B_1 + A_10, A_0^2 + A_0, A_3^2 + A_6, A_3^4 + A_5, A_0 + A_7
, A_1 + A_8, A_2 + A_9, A_3 + A_10, A_1, A_2, A_4}

o27 : List
```

Hence, using computations in Macaulay2, one can construct system $\eta_C(w)$ of polynomial equations in several variables. Now, in order to determine whether this system has a solution or not, one need to construct an ideal generated by polynomial equations of this system in a multivariable polynomial ring and compute its Gröbner basis. But it is always better to provide the simplified form of above system by substituting expressions of the type $A_{iq \bmod n} = A_i^q$, $A_{i+n} = A_i$ and $A_i = 0, \forall i \in I(\mathcal{C})$ in polynomial equations of the type $A_{i+w} + \sigma_1 A_{i+w-1} + \cdots + \sigma_w A_i = 0$. For example, the simplified form of system $\eta_C(3)$ is as follows:

$$\eta_{\mathcal{C}}(3) = \begin{cases} A_3\sigma_1 = 0 \\ A_3^4 + A_3^3\sigma_2 = 0 \\ A_3^2 + A_3^4\sigma_1 + A_3^3\sigma_3 = 0 \\ A_0 + A_3^2\sigma_2 + A_3^4\sigma_4 = 0 \\ A_0\sigma_1 + A_3\sigma_2 + A_3\sigma_3 = 0 \\ A_0\sigma_2 + A_3^2\sigma_3 = 0 \\ A_3 + A_o\sigma_3 = 0 \\ A_0^2 = A_0 \end{cases}$$

Following user defined function is constructed in macaulay2, which simplifies the system as discussed.

```
i28 : system = (l,q,n,w) ->

      (
         g'={};
      for j from (w+1) to (w+n) do
      (
            l1={};
            for i from 1 to (w+1) do
            (
              b= ((j+1)-i)%n;
              if (isMember(b,defining_set(l,q,n))) then l1=join(l1,{0})
              else
              (
                    p=0;
                    l'= lcc(q,n); k={};
                    for m from 0 to (length(l')-1) do
                    (
                        h=l'_m_0;
                        if (isMember(h,defining_set(l,q,n))) then continue else k=join(k,{h})
                    );
                    for m from 0 to (length(k)-1) do
                    (
                        if (isMember(b,cyclotomic_coset(k_m,q,n))) then (p=k_m; s= cyclotomic_coset(p,q,n); break)
                    );
                    for e from 0 to (length(s)-1) do
                    (
                       if (b == s_e) then l1=join(l1,{((A_p)^(q^e))})
                    )
              )
            );
            l2={1};
            for i from 1 to w do(
              l2=join(l2,{B_i}));
            f1=0;
            for i from 0 to w do(
              f1=f1+(l1_i * l2_i));
            g'=join(g',{f1})
      );
      g'
      )

o28 = system

o28 : FunctionClosure
```

```
i29 : system({1},2,7,3)

o29 = {A_3*B_1, A_3^4+A_3*B_2, A_3^4*B_1+A_3^2+A_3*B_3, A_0+A_3^4*B_2+A_3^2*B_1, A_0*B_1+A_3^4*B_3+A_3^2*B_2, A_0*B_2+A_3^2*B_3, A_0*B_3+A_3}

o29 : List
```

Following is one more demonstration for constructing the simplified form of the algebraic system $\eta_{\mathcal{C}}(6)$ where $\mathcal{C}$ is a cyclic code of length 63 over $\mathbb{F}_2$ with defining set, $cl(1) \cup cl(5) \cup cl(7) \cup cl(9) \cup cl(11) \cup cl(13) \cup cl(23) \cup cl(27)$

```
i31 : R=GF(2)[A_0,A_1..A_69,B_1..B_6,MonomialOrder=>Lex]

o31 = R

o31 : PolynomialRing

i32 : system({1,5,7,9,11,13,23,27},2,63,6)
```

$$o32 = \{A_3^2B_1 + A_3B_4,\ A_3^2B_2 + A_3B_5,\ A_3^2B_3 + A_3B_6,\ A_3^2B_4,\ A_3^2B_5,\ A_3^4 + A_3^2B_6,\ A_3^4B_1,\ A_3^4B_2,\ A_3^4B_3 + A_{15},\ A_3^4B_4 + A_{15}B_1,\ A_3^4B_5 + A_{15}B_2,\ A_3^4B_6 + A_{15}B_3,\ A_{15}B_4,\ A_{15}B_5,\ A_{15}B_6 + A_{21},\ A_{21}B_1,\ A_{21}B_2,\ A_3^8 + A_{21}B_3,\ A_3^8B_1 + A_{21}B_4,\ A_3^8B_2 + A_{21}B_5,\ A_3^8B_3 + A_{21}B_6,\ A_3^8B_4,\ A_3^8B_5,\ A_3^8B_6 + A_{15}^2,\ A_{15}^2B_1 + A_{31},\ A_{15}^2B_2 + A_{31}B_1,\ A_3^{32} + A_{15}^2B_3 + A_{31}B_2,\ A_3^{32}B_1 + A_{15}^2B_4 + A_{31}B_3,\ A_3^{32}B_2 + A_{15}^2B_5 + A_{31}B_4,\ A_3^{32}B_3 + A_{15}^2B_6 + A_{31}B_5,\ A_3^{32}B_4 + A_{31}B_6,\ A_3^{32}B_5,\ A_3^{32}B_6 + A_{15}^{32},\ A_{15}^{32}B_1,\ A_{15}^{32}B_2,\ A_{15}^{32}B_3 + A_{21}^2,\ A_{15}^{32}B_4 + A_{21}^2B_1,\ A_{15}^{32}B_5 + A_{21}^2B_2,\ A_{15}^{32}B_6 + A_{21}^2B_3,\ A_{21}^2B_4,\ A_{21}^2B_5 + A_{31}^{32},\ A_3^{16} + A_{21}^2B_6 + A_{31}^{32}B_1,\ A_3^{16}B_1 + A_{31}^{32}B_2,\ A_3^{16}B_2 + A_{31}^{32}B_3,\ A_3^{16}B_3 + A_{15}^{16} + A_{31}^{32}B_4,\ A_3^{16}B_4 + A_{15}^{16}B_1 + A_{31}^{32}B_5,\ A_3^{16}B_5 + A_{15}^{16}B_2 + A_{31}^{32}B_6,\ A_3^{16}B_6 + A_{15}^{16}B_3,\ A_{15}^{16}B_4 + A_{31}^{16},\ A_{15}^{16}B_5 + A_{31}^{16}B_1,\ A_{15}^{16}B_6 + A_{15}^8 + A_{31}^{16}B_2,\ A_{15}^8B_1 + A_{31}^{16}B_3,\ A_{15}^8B_2 + A_{31}^{16}B_4 + A_{31}^8,\ A_{15}^8B_3 + A_{15}^4 + A_{31}^{16}B_5 + A_{31}^8B_1,\ A_{15}^8B_4 + A_{15}^4B_1 + A_{31}^{16}B_6 + A_{31}^8B_2 + A_{31}^4,\ A_{15}^8B_5 + A_{15}^4B_2 + A_{31}^8B_3 + A_{31}^4B_1 + A_{31}^2,\ A_0 + A_{15}^8B_6 + A_{15}^4B_3 + A_{31}^8B_4 + A_{31}^4B_2 + A_{31}^2B_1,\ A_0B_1 + A_{15}^4B_4 + A_{31}^8B_5 + A_{31}^4B_3 + A_{31}^2B_2,\ A_0B_2 + A_{15}^4B_5 + A_{31}^8B_6 + A_{31}^4B_4 + A_{31}^2B_3,\ A_0B_3 + A_3 + A_{15}^4B_6 + A_{31}^4B_5 + A_{31}^2B_4,\ A_0B_4 + A_3B_1 + A_{31}^4B_6 + A_{31}^2B_5,\ A_0B_5 + A_3B_2 + A_{31}^2B_6,\ A_0B_6 + A_3^2 + A_3B_3\}$$

```
o32 : List
```

Hence, For a cyclic code C of length $'n'$ over $\mathbb{F}_q$, with BCH bound $'w'$, this simplified form of system $\eta_C(w)$ which is constructed in Macaulay2, can be used to construct an ideal I in a polynomial ring of several variables, generated by polynomial equations of the simplified system and Gröbner basis of that ideal can be computed using available Macaulay2 commands. If the minimal Gröbner basis of this ideal does not reduces to $\{1\}$ then the system $\eta_C(w)$ has a solution (see [3]) and that guarantees existence of a codeword in C of weight $'w'$. Number of solutions of $\eta_C(w)$ will determine the number of codewords in C of weight $'w'$. Hence, constructing such systems in Macaulay2 are of great help to describe minimum weight codewords of a cyclic code.

REFERENCES

1. S. Ling, C. Xing, “Coding Theory-A first Course”, Cambridge university press, New York, 2004, pp.133-187.
2. I. Cascudo, "On Squares of Cyclic Codes," *IEEE Transactions on Information Theory*, vol. 65, no. 2, pp. 1034-1047, Feb. 2019
3. D.A. Cox, J.Little and D O’shea, “Ideals, earieties and Algorithms – An introduction to Computational Algebraic Geometry and Commutative Algebra”, 4th ed., Springer, 2015, pp. 1-120.
4. F.J.MacWilliams, N.J.A.Sloane, “Theory of error correcting codes”, North-Holland Publishing Company, New York,1977, pp. 1- 293
5. D. Augot, “Description of Minimum Weight Codewords of Cyclic Codes by Algebraic Systems”, Finite Fields and Their Applications, eolume 2, Issue 2, 1996, pp.138-152.
6. D. Augot, P. Charpin and N. Sendrier, "Studying the locator polynomials of minimum weight codewords of BCH codes," *IEEE Transactions on Information Theory*, vol. 38, no. 3, May 1992, pp. 960-973
7. H. Liu, X. Wang, D. Zheng, “On the weight distributions of a class of cyclic codes”, Discrete Mathematics, eolume 341, Issue 3, 2018, Pages 759-771