CHAPTER 4
PROJECTIVE GEOMETRY &
PROJECTIVE GEOMETRY CODES

4.1 INTRODUCTION

The higher girth in the tanner graph [62],[65] allows more accurate computation in the information exchange of belief propagation method of decoding and hence a better performance is obtained using the concept of projective geometry. PG-LDPC codes can be constructed using the incidence relation between lines and points of PG over finite fields. A brief description of concept of projective geometry in the context of code construction is discussed below.

Projective Geometry [2],[62],[71] is constructed from the elements of a Galois field. Consider the Galois field \( \text{GF}(2^{(m+1)s}) \) of m tuples containing \( \text{GF}(2^s) \) as a subfield. Assuming ‘\( \lambda \)’ as primitive element, belonging to \( \text{GF}(2^{(m+1)s}) \), one can find \( \lambda^1, \lambda^2, \lambda^3, \ldots, \lambda^{2^s(m+1)s-2} \) elements which are non zero. Consider the ratio \( \rho \) which can be used as power of primitive element \( \lambda \)

\[
\rho = \frac{2^{m+1}s-1}{2^s-1} = 2^{m+1}s + 2^{(m+1s)-1} + \ldots + 2^{s} + 1 \quad (4.1)
\]

Then the order of the factor ‘\( \gamma \)’ is \( \rho \)th power of \( \lambda = \lambda^\rho \) is \( 2^{s-1} \). The \( 2^s \) elements 0,1,\( \gamma, \gamma^2, \ldots, \gamma^{2^{s-2}} \) as given in [2] form the Galois field \( \text{GF}(2^s) \).

4.2 POINTS IN PROJECTIVE GEOMETRY

The first \( \rho \) powers of \( \lambda \) is the set

\[
V = \{ \lambda^0, \lambda^1, \lambda^2, \ldots, \lambda^{2^{s-1}} \} \quad (4.2)
\]
No element $\lambda^k$ in $V$ exists as a multiple of an element in GF($2^s$) with another element in $V$ i.e., $\lambda^k$ is not equal to $0\lambda^i$ for $\delta$ which belongs to GF($2^s$). If $\lambda^k \equiv 0 \lambda^i$, implies $\lambda^k = \delta \lambda^i$. Since $\delta^{2^k - 1} = 1$, $\lambda^{(k-l)(2^s-1)} = 1$ which is impossible because $(k-l)(2^s-1) < (2^{(m+1)s} - 1)$ while the order of $\lambda$ is $2^{(m+1)s} - 1$. Hence, $\lambda^k \neq 0 \lambda^i$ for $\delta \in$ GF($2^s$). If $\gamma$ is the primitive element belonging to the Galois Field GF($2^s$), partition other than zero elements of GF($2^{(m+1)s}$) into $\rho$ disjoint subsets as follows:

$$\{\lambda^i, \gamma\lambda^i, \gamma^2\lambda^i, \ldots, \gamma^{2^{s-2}}\lambda^i\} \quad \text{for } 0 \leq i < \rho$$

There are $2^s - 1$ elements in every set and every element satisfies the condition of being product of the first element in the set. Therefore, each set can be represented by its first element as

$$(\lambda^k) = (\lambda^k, \gamma \lambda^k, \gamma^2 \lambda^k, \ldots, \gamma^{2^{s-2}} \lambda^k) \quad \text{where } 0 \leq k < \rho \quad \ldots \quad (4.3)$$

For any $\lambda^i$ in GF($2^{(m+1)s}$), if $\lambda^i = \gamma^p \lambda^k$ with $0 \leq p < \rho$, then $\lambda^i$ is in $(\lambda^k)$. The $\rho$ elements given by $(\lambda^0), (\lambda^1), (\lambda^2), \ldots, (\lambda^{\rho-1})$ are said to form an $m$-dimensional projective geometry over the Galois field GF($2^s$) denoted as PG($m$, $2^s$). The elements $(\lambda^0), (\lambda^1), (\lambda^2), \ldots, (\lambda^{\rho-1})$ are known as points of PG($m$, $2^s$).

**4.3 LINES IN PROJECTIVE GEOMETRY**

Assume $(\lambda^k), (\lambda^l)$ are the two distinct points in PG($m$, $2^s$). Then line passing through the above two points, consists of points of the form given by
There are \( (2^s)^2 - 1 \) different possible choices of \( \sigma_1 \), \( \sigma_2 \) from \( \text{GF}(2^s) \) (excluding \( \sigma_1 = \sigma_2 = 0 \)). However, there are always \( 2^s - 1 \) choices of \( \sigma_1 \) and \( \sigma_2 \) that result in the same point in \( \text{PG}(2^{(m+1)s}) \) i.e.,

\[
\sigma_1 \lambda^i + \sigma_2 \lambda^j, \quad \gamma \sigma_1 \lambda^i + \gamma \sigma_2 \lambda^j, \quad \ldots \quad \gamma^{2^s - 2} \sigma_1 \lambda^i + \gamma^{2^s - 2} \sigma_2 \lambda^j
\]

represent the same point in \( \text{PG}(2^{(m+1)s}) \). A line in \( \text{PG}(m, 2^s) \) therefore consists of

\[
\frac{(2^s)^2 - 1}{2^s - 1} = 2^s + 1 \text{ points.}
\]

In order to generate \( 2^s + 1 \) distinct points on the line \( \{ \sigma_1 \lambda^i + \sigma_2 \lambda^j \} \), choose \( \sigma_1 \) and \( \sigma_2 \) such that no choice of \( \{ \sigma_1, \sigma_2 \} \) is a multiple of another choice of \( \{ \sigma_1, \sigma_2 \} \) [i.e., \( \{ \sigma_1, \sigma_2 \} \neq \{ \delta \sigma_1, \delta \sigma_2 \} \) for \( \delta \in \text{GF}(2^s) \)].

**Illustration:** For \( m=s=2 \), consider \( \text{PG}(2, 2^2) \). This geometry can be constructed from the field \( \text{GF}(2^6) \), containing the subfield \( \text{GF}(2^2) \) with the number of points equal to

\[
\frac{2^{(m+1)s} - 1}{2^s - 1} = 21
\]

Assume \( \lambda \) to be primitive element of \( \text{GF}(2^6) \). Assume \( \gamma = \lambda^{21} \). Then \( (0,1,\gamma,\gamma^2) \) form the subfield \( \text{GF}(2^2) \). \( \text{PG}(2, 2^2) \) contains 21 points from \( (\lambda^0) \) to \( (\lambda^{20}) \).

Consider a line passing through the point \( (\lambda^1) \) and \( (\lambda^{20}) \) consisting of 5 points of the form \( (\sigma_1 \lambda^1 + \sigma_2 \lambda^{20}) \) with \( \sigma_1, \sigma_2 \) from \( \text{GF}(2^2) = \{0,1,\gamma,\gamma^2\} \). The 5 distinct points are

\[
(\lambda^1), \quad (\lambda^{20}).
\]
\[ (\lambda^1 + \lambda^{20}) = (\lambda^5) = y^2 \lambda^{45} = (\lambda^{14}), \]
\[ (\lambda^1 + y\lambda^{20}) = (\lambda^3 + \lambda^{41}) = (\lambda^{65}) = y^2 \lambda^{36} = (\lambda^{14}), \]
\[ (\lambda^1 + y^2\lambda^{20}) = (\lambda^1 + \lambda^{62}) = (\lambda^{11}) \]

Therefore, \{ (\lambda^1), (\lambda^{11}), \lambda^{14}, \lambda^{15}, (\lambda^{20}) \} forms a line in PG(2, 2^2) which passes through the points \( \lambda^1 \) and \( \lambda^{20} \). This line passes through points 1, 11, 14, 15, 20 is shown in the fig. 4.1.

![A LINE IN PG (2,2^2)](image)

Fig. 4.1 Line in PG(2,2^2)

### 4.4 INTERSECTION OF 2 LINES IN PG(m, 2^n)

Consider two lines represented by \( \{(\sigma_1 \lambda^k + \sigma_2 \lambda^i)\} \) and \( \{(\sigma_1 \lambda^k + \sigma_2 \lambda^j)\} \). Let point \( \lambda^i \) be not on the line \( \{(\sigma_1 \lambda^k + \sigma_2 \lambda^i)\} \). It implies that two lines do have only one common point \( \lambda^i \) which means that they are intersecting at the \( \lambda^i \) common point.

The lines that meet at a known point are given by

\[
\frac{2^{(m-1)} - 1}{2^{m-1}} = 1 + 2^2 + 2^{(m-3)}w + \ldots + 2^{(m-1)}w 
\]

\( (4.5) \)
There are \( J = (2^{ms+\ldots+2^s+1})(2^{(m-1)s+\ldots+2^s+1})/(2^s-1) \) lines in \( \text{PG}(m, 2^s) \).

4.5 DEFINITION OF INCIDENCE MATRIX

The Incidence vector of a line is given in \( \text{PG}(m, 2^s) \) as the n-tuple form

\[
V = (v_0, v_1, v_2, \ldots, v_{n-1}) \tag{4.6}
\]

whose \( i^{th} \) component \( v_i = 1 \) if \( (\lambda^i) \) is a point on the line of \( \text{PG}(m, 2^s) \), and 0 otherwise.

Value of \( n = (2^{(m+1)s-1})/(2^s-1) = 2^{ms+2^{(m-1)s+\ldots+2^s+1}}. \)

The matrix formed by the incidence vectors belonging to the set of all lines in \( \text{PG}(m, 2^s) \) is called the incidence matrix of \( \text{PG}(m, 2^s) \). Number of rows of the matrix is equal to number of lines in \( \text{PG}(m, 2^s) \) and number of columns is equal to number of points on \( \text{PG}(m, 2^s) \).

Some structural properties of Incidence Matrix are:

1. Each row contains the weight = \( 2^s+1 \)

2. Each column contains the weight = \( (2^{ms-1})/(2^s-1) \)

3. There will be only “1-component” in common between any two columns.

4. On the similar lines, any two rows have at most only one “1-component” in common.
5. The density of Incidence Matrix = \((2^{2s}-1)/(2^{(m+1)s}-1)\). For \(m\geq 2\), \(r\) is relatively small. Since the incidence matrix is sparse in nature, it is called as the parity check matrix \(H\).

### 4.6 PG-LDPC Code Definition

A binary PG code \(C\) over \(PG(m, 2^s)\) of length \(n= (2^{(m+1)s}-1)/(2^s-1) = 2^{ms}+2^{(m-1)s}+\ldots.+2^s+1\) is defined as the largest cyclic code whose null space is the incidence matrix \(H\) of \(PG(m, 2^s)\). Since column weight of \(H\) is \((2^{ms}-1)/(2^s-1)\), the minimum distance of \(C = (2^{ms}-1)/(2^s-1)+1\). Since the \(H\) matrix is sparse, the code \(C\) is called PG-LDPC codes. The properties of these codes lends them to be decoded using majority logic decoding and at the same time being cyclic, generation of codes is rather easy. Therefore they can be encoded with a linear feedback shift register based on generator polynomial.

**Definition of \(W_{2^s}(d)\) (i.e., \(2^s\)-weight of \(d\)):**

Let \(d\) be a non negative integer < \(2^{(m+1)s}-1\). \(d\) can be expressed in radix \(2^s\) form given in \([2]\) as follows

\[
d = \delta_0 + \delta_1 2^s + \delta_1 2^{2s} + \ldots. + \delta_m 2^{ms} \quad \text{where} \quad 0 \leq \delta_i < 2^s
\]

define \(2^s\)-weight of \(d\) (denoted \(W_{2^s}(d)\)) as the following sum

\[
W_{2^s}(d) = \delta_0 + \delta_1 + \delta_2 + \ldots. + \delta_m
\]
4.7 ROOTS OF GENERATOR POLYNOMIAL

Let $d$ be a non negative integer $< 2^{(m+1)s} - 1$. Let $d^{(l)}$ be the remainder obtained by dividing $2^d$ by $2^{(m+0)s} - 1$. The generator polynomial $g(X)$ of PG code over $(m, 2^s)$ of length $\rho = \frac{2^{ms} - 1}{2^s - 1}$ has as a root if it is divisible by $2^s - 1$ and

$$0 \leq \max_{0 \leq d \leq 2^s} (d^{(l)}) = j(2^s - 1) \text{ with } 0 \leq j \leq m-1 \quad (4.7)$$

From the characterization of roots of $g(X)$ given by (4.7) it can be shown that $g(X)$ has following consecutive powers of $\zeta = \lambda^{(2^s - 1)}$:

$$\zeta^0, \zeta^1, \zeta^2, \zeta^3, \ldots, \zeta^{(2^{ms} - 1)/(2^s - 1)} \quad (4.8)$$

Therefore from BCH bound minimum distance of C is $(2^{ms} - 1)/(2^s - 1) + 1$

4.8 TWO-DIMENSIONAL PG-LDPC CODES

A special subclass of PG-LDPC codes is the class of PG-LDPC codes over PG(2,2^s) for various $s$ and $m=2$. For any $s \geq 2$ the 2-D PG-LDPC code's parameters are:

- **Length** $= n = 2^{2s+2^s+1}$
- **Number of parity bits** = $(n-k) = 3^s+1$
- **Number of information bits** = $k = n-(3^s+1)$
- **Minimum distance** $= 2^s+2$
- **Row weight of H matrix** $= 2^s+1$
- **Column weight of H matrix** $= 2^s+1$
The H matrix is of dimension \((2^{2s}+2^s+1)\)-by- \((2^{2s}+2^s+1)\). It can be obtained by taking the incidence vector of a line in PG(2, \(2^s\)) and its \(2^{2s}+2^s\) shifts.

Table 4.1  A List of 2D PG-LDPC Codes

<table>
<thead>
<tr>
<th>S</th>
<th>N</th>
<th>K</th>
<th>Dmin</th>
<th>Row weight</th>
<th>Column weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>21</td>
<td>11</td>
<td>6</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>73</td>
<td>45</td>
<td>10</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>273</td>
<td>191</td>
<td>18</td>
<td>17</td>
<td>17</td>
</tr>
<tr>
<td>5</td>
<td>1057</td>
<td>813</td>
<td>34</td>
<td>33</td>
<td>33</td>
</tr>
<tr>
<td>6</td>
<td>4161</td>
<td>3431</td>
<td>66</td>
<td>65</td>
<td>65</td>
</tr>
<tr>
<td>7</td>
<td>16513</td>
<td>14326</td>
<td>130</td>
<td>129</td>
<td>129</td>
</tr>
</tbody>
</table>

4.9 THE FLOW CHART AND ALGORITHM FOR DESIGNING 2-D PG CODES (OVER PG(2, \(2^s\)))

A generalized approach for the construction of PG codes is given below initially with a flow chart and followed by an algorithmic approach.
Flow chart of 2DPG:

Start

Enter the values of s, p

Find code length, info. length, row wt, col.wt.

Find degree of GF, primitive polynomial

Generate multiplicative group of order $p^{\text{degree}}-1$ over $\mathbb{Z}_2$

Obtain feedback polynomial over $\mathbb{Z}_2$

Generate GR,GF using feedback polynomial

Generate log and antilog tables

Generate set of lines

Generate Incidence Matrix over GF(2), H Matrix over $\mathbb{Z}_2$

Find density of the Matrix, dual Matrix

Generate Code Vector

Output info. vector, code

Stop

Fig. 4.2 Flow chart for generation of 2D PG
The construction of the 2D LDPC Codes consists of the generation of

- Primitive polynomial,
- Log and Anti log tables
- Points, Lines
- Incidence matrix,
- Generator and parity check matrix

The algorithm ‘2DPG’ is tested using Matlab software where in the comments provide required clarity at different points and hence the algorithm is self explanatory and consists of following steps.

**Step 1** For the given degree “s” of the subfield GF(2^s), generate GF(2^{3s}) using LFSR sequence generator with the feedback connections(feedback) decided by the primitive polynomial(prim_poly)

**Step 2** Generate finite field Log and Antilog tables (ZAK Tables) to facilitate Galois Field arithmetic

**Step 3** Define the 2^{2s}+2^s+1 consecutive rows of the GF(2^{3s}) as points(PointsOfPG) of the PG(2, 2^s). The points are labelled from 0 to 2^{2s}+2^s. Point “j” is given by the row “j” of GF(2^{3s}). Label of Point “j” implies it is the point λ_i of GF(2^{3s}).

**Step 4** Define the base line as line 0 and obtain it as the line joining the points “0” and “1”. A total of 2^s+ 1 points are present on the line as given below:
Point 0 = \( \lambda^0 \);

Point 1 = \( \lambda^1 \);

Point 2 = \( \lambda^0 + \gamma^1 \);

Point 3 = \( \lambda^0 + \gamma^s \lambda^1 \);

Point 4 = \( \lambda^0 + \gamma^{2s} \lambda^1 \);

Point \( 2^s = \lambda^0 + \gamma^{(2^s-2)s}\lambda^1 \);

Here \( \gamma \) is the primitive element of the subfield

If line “0” = (p0, p1, p2, p3, ……p\( 2^s \)), the rest of the lines are obtained as Line “i” = (p0+i, p1+i, p2+i, p3+i, ……….p\( 2^s+i \)) mod (\( 2^{2^s+2s+1} \))

for \( i = 1 \) to \( 2^{2^s+2s} \).

**Step 6** Compute the subfield elements 0, 1, \( \gamma \), \( \gamma^2 \), \( \gamma^3 \)…… \( \gamma^{2^s-2} \) with \( \gamma = \lambda^{(2^s(2^s)+2^s+1)} \).

**Step 7** The generator polynomial is computed using the following procedure:

1. Find out the allowable set (AllowedSet). Allowable Set is the set of all integers < \( 2^{3s} \) divisible by \( (2^s-1) \)

2. Find out the j-set (jSet). It is the set of elements \{ j(2^s-1) \} for 0\( \leq j \leq m-1 \).
3. Find the roots (roots) of $g(X)$ by implementing equation (4.7) using the results of sub-steps 1 and 2

**Step 8** Using the roots information obtained in step 7 compute $g(X)$ using convolution over Galois Field involving two further sub steps. One can use Matlab communication Tool-box facility for implementing Step 8.

1. Convert binary vectors representing $GF(2^{3s})$ elements into decimal values($GF2Dec$) for use in sub-step2

2. Convolve the roots using functions of Communication tool box to get $g(X)$

**Step 9** Get the generator matrix (G_MATRIX). The first row contains an n-tuple whose first n-k+1 elements are the coefficients of $g(X)$ with LSB first. The rest of the rows are obtained by shifting the first row k-1 times.

**Step 10** The parity check matrix which is known as the incidence matrix is obtained.

It is a $(2^{2s+2s+1})$-by-$(2^{2s+2s+1})$ binary matrix

**Pseudocode of 2DPG**

- $s = \text{input('Enter 's',Degree of the Subfield)}$
- $p = 2;$
- $\text{disp('Dimension of the Projective Space')}$
- $m = 2$
Degree of the Galois Field is $GF(2^{3s})$:

- $\text{DegreeOfGF} = (m+1)s$
- $\text{degree} = \text{DegreeOfGF}$;

Size of the Galois Field $GF(2^{3s})$:

- $\text{SizeOfGF} = 2^{((m+1)s)}$

**STEP1:**

Construct Galois Field $(2^{3s})$:

*Note: prim_poly is Primitive Polynomial. Use lookup table*

- if (degree == 3)
  
  ```
  prim_poly = [1 0 1 1];
  ```

- elseif (degree == 4)
  
  ```
  prim_poly = [1 1 0 0 1];
  ```

- elseif (degree == 5)
  
  ```
  prim_poly = [1 0 1 0 0 1];
  ```

- elseif (degree == 6)
  
  ```
  prim_poly = [1 1 0 0 0 1];
  ```

- elseif (degree == 7)
  
  ```
  prim_poly = [1 0 0 1 0 0 1];
  ```

- elseif (degree == 8)
  
  ```
  prim_poly = [1 0 1 1 0 0 0 1];
  ```

- elseif (degree == 9)
  
  ```
  prim_poly = [1 0 0 0 1 0 0 0 1];
  ```

- elseif (degree == 10)
  
  ```
  prim_poly = [1 0 0 1 0 0 0 0 1];
  ```
elseif(degree == 11)
    prim_poly = [1 0 1 0 0 0 0 0 0 0 1];
elseif(degree == 12)
    prim_poly = [1 1 0 0 1 0 1 0 0 0 0 1];
elseif(degree == 15)
    prim_poly = [1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
elseif(degree == 18)
    prim_poly = [1 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1];
elseif(degree == 21)
    prim_poly = [1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1];
end

Prim_Poly = prim_poly

Getting Feedback Taps:

- flipped_prim_poly = fliplr(prim_poly);
- feedback = flipped_prim_poly(2:degree+1);
- FeedbkTaps = feedback

Generate non-zero elements of GF:

- NoOfNonZeroElements = p^degree-1;
- shiftreg = zeros(1, degree);
- shiftreg(degree) = 1;
- GF = zeros(NoOfNonZeroElements, degree+1);
- for i = 1:NoOfNonZeroElements
      carry = shiftreg(1);
      temp = shiftreg(2:degree);
shiftreg = [temp 0];
shiftreg = shiftreg + carry*(feedback);
shiftreg = rem(shiftreg, p);
GF(i, 1:degree+1) = [i shiftreg];

• end

Introduction of all zero field element:

• GF(NoOfNonZeroElements+1, 1:degree+1) =
  [NoOfNonZeroElements+1 zeros(1, degree)];

Rearrangement for displaying galoisfield:

• temp = [GF(NoOfNonZeroElements, :)

• GF(1:NoOfNonZeroElements-1, :)];

• GaloisField = temp;

• GaloisField(1, 1) = 0

STEP2:

Generating log and antilog tables:

Generating log table:

• zek = zeros(length(GF(:, 1)), 2);

• for i = 1:length(GF(:, 1))-2

    temp = GF(i, 1:degree+1);

    acc = 0;

    for j = 1:degree

        acc = acc + temp(j+1)*p^(degree-j);

    end

    intval = acc;
\[ \text{zek}(i, 1:2) = [\text{temp}(1) \text{ intval}]; \]

- end
- \[ \text{zek(NoOfNonZeroElements, 1:2)} = [\text{NoOfNonZeroElements} 1]; \]
- \[ \text{zek(NoOfNonZeroElements+1, 1:2)} = [\text{NoOfNonZeroElements}+1 0]; \]
- \[ \logtbl = \text{zek}; \]

**Generating Antilog Table**
- \[ \logval = \logtbl(:, 2); \]
- \[ [Y, I] = \text{sort}(\logval); \]
- \[ \text{antilogtbl} = [Y I]; \]

**Number Of Points On 2-D Projective Space**
- \[ \text{noofpoints} = 2^{(2s)}+2^s+1; \]
- \[ \text{NoOfPointsOnPG} = \text{noofpoints} \]

**STEP3:**

**Points Of The 2 Dimensional Projective Space**

- \[ \text{PointsOfPG} = [\text{GF(NoOfNonZeroElements, :)} \]
- \[ \text{GF([1:noofpoints-1, :])}; \]
- \[ \text{PointsOfPG(1, 1)} = 0 \]

**STEP4:**

**Generating Baseline Joining Point "0" and Point "1"**

- \[ \text{gamma} = 2^{(2s)}+2^s+1; \]
- \[ \text{Label Of Primitive Element Of GF}(2^S) \]
- \[ \text{cnt} = 1; \]
Define Point "0"

- \( pt0 = \text{zeros}(1, \text{degree}-1) \ 1 \);
- \( \text{baseline}(\text{cnt}) = 0; \) % Label of the point
- \( \text{cnt} = \text{cnt} + 1; \)

Define Point "1"

- \( pt1 = \text{zeros}(1, \text{degree}-2) \ 1 \ 0 \);
- \( \text{baseline}(\text{cnt}) = 1; \)
- \( \text{cnt} = \text{cnt} + 1; \)

For Beta = 1

- \( \text{temp} = \text{rem}(pt0 + pt1, 2); \)
- \( \text{acc} = 0; \)
- for \( j = 1: \text{degree} \)
  
  \( \text{acc} = \text{temp}(j) \cdot p^{(\text{degree}-j)} + \text{acc}; \)

end

- \( \text{intval} = \text{acc}; \)
- \( \text{power} = \text{antilogtbl}(\text{intval}+1, 2); \)
- \( \text{baseline}(\text{cnt}) = \text{rem}(\text{power}, \gamma); \)
- \( \text{cnt} = \text{cnt} + 1; \)

For gamma = 1:2^s-2

- for \( i = 1:2^s-2 \)
  
  \( \text{temp} = \text{rem}(pt0 + \text{GF}(i*\gamma+1, 2: \text{degree}+1), 2); \)

  \( \text{acc} = 0; \)

  for \( j = 1: \text{degree} \)
  
  \( \text{acc} = \text{temp}(j) \cdot p^{(\text{degree}-j)} + \text{acc}; \)
end
intval = acc;
power = antilogtbl(intval+1, 2);
baseline(cnt) = rem(power, gamma);
cnt = cnt + 1;

• end
• baseline = sort(baseline);
• Line0 = baseline

**STEP 5:**

*Generating Lines*

• disp('Lines of PG: Line Construction Technique Is Due To Singer')
• lines = [];
• lines(1, :) = baseline;
• ref = baseline;
• for i = 2:gamma
  
  ref = rem(ref + ones(1, p^s+1), gamma);

  lines(i, :) = ref;
• end
• LinesOfPG = lines

**STEP 6:**

*Elements Of Subfield*

• SubFieldElements = zeros(2^s, degree+1);
• SubFieldElements(1, :) = [0 zeros(1, degree)];
• SubFieldElements(2, :) = [1 zeros(1, degree-1) 1];
• for i = 2:2^s-1
    SubFieldElements(i+1, :) = [i GF((i-1)*gamma,2:degree+1)];
• end
• disp('Subfield Elements')
• SubFieldElements

**STEP7:**

*Finding roots of G(x) polynomial*

**Substep 1**

• disp('Allowed Set Condition 1')
• AllowedSet = [];
• cnt = 1;
• for i = 0:2^((m+1)*s)-2
    if(rem(i, 2^s-1) == 0)
        AllowedSet(cnt) = i;
        cnt = cnt + 1;
    end
    end
• AllowedSet

**Substep 2**

• jSet = [];
• for j = 0:m-1
    jSet(j+1) = j*(2^s-1);
• end
disp('jSet')
• jSet

*Substep3*

• disp('The Roots')
• roots = [];
• cnt = 1;
• for i = 1:length(AllowedSet)
    h = AllowedSet(i);
    hl = 0;
    wt_of_h = 0;
    for l = 0:s-1;
        hl = rem(2^l*h, 2^((m+1)*s)-1);
        temp = hl;
        delta = 0;
        for j = 1:m+1
            a = rem(temp, 2^s);
            delta(j) = a;
            temp = floor(temp/2^s);
        end
        hl(l+1, 1:m+1) = delta;
        wt_of_h(l+1) = sum(delta);
    end
• flg = 0;
    for j = 1:length(jSet)
        if(max(wt_of_h) == jSet(j))
\[ \text{flg(j)} = 1; \]

\[ \text{else} \]
\[ \text{flg(j)} = 0; \]
\[ \text{end} \]

- \[ \text{end} \]
- \[ \text{if}(\text{sum(flg)} \geq 1) \]
- \[ \text{roots(cnt)} = \text{AllowedSet(i)}; \]
- \[ \text{cnt} = \text{cnt}+1; \]
- \[ \text{end} \]
- \[ \text{end} \]
- \[ \text{roots} \]

**STEP 8:**

*Finding G(x) Using "Roots" Information*

*Use Facility Available In Matlab Communication Tool Box To Get G(x)*

*Substep 1*

*GF To Decimal Conversion To Facilitate Convolution*

- \[ \text{GF2dec} = []; \]
- \[ \text{for} \ i = 1: \text{length(GF(:, 1))}-1 \]
- \[ \text{ref} = \text{GF}(i, 2: \text{degree}+1); \]
- \[ \text{ref} = \text{fliplr(ref)}; \]
- \[ \text{acc} = 0; \]
- \[ \text{for} \ j = 1: \text{degree} \]
  \[ \text{acc} = \text{acc}+\text{ref(j)}*2^\text{j-1}; \]
- \[ \text{end} \]
\begin{itemize}
\item if(i ~= 2^{\text{degree}-1})
\item \text{GF2dec}(i, \text{1:2}) = [i \text{ acc}];
\item else
\item \text{GF\_dec}(i, \text{1:2}) = [0 \text{ acc}];
\item end
\item end
\end{itemize}

\textit{Substep 2}

\textit{Convolution In Galois Field}

\begin{itemize}
\item \(\text{gx} = [1 \ \text{gf}(1, \ \text{degree})] ;\)
\item for \(i = 2: \text{length(roots)}\)
\begin{itemize}
\item \(\text{ref2} = \text{gf} \left( \text{GF2dec} \left( \text{roots}(i), \text{2} \right) , \text{degree} \right) ;\)
\item \(\text{gx} = \text{conv}([1 \ \text{ref2}], \text{gx});\)
\end{itemize}
\item end
\item \(\text{gx} = \text{fliplr}(\text{gx});\)
\item \(\text{disp('gx = 1 +a(1)x+a(2)x^2+a(3)x^3+\ldots\ldots\ldots+a(n-k)x^{(n-k)\prime'})}\)
\item \(\text{gx} = \text{double(} \text{gx} == 1)\)
\end{itemize}

\textbf{STEP 9:}

\textit{Get G\_Matrix}

\begin{itemize}
\item \(\text{disp('k= \ The length of information vector is equal to total length -}3^\text{S+1}\prime)\)
\item \(n = 2^{(2^s) s}+2^s+1\)
\item \(k = n-(3^s+1)\)
\item \(\text{disp('G MATRIX')}\)
\end{itemize}
• G_MATRIX = [];
• g = zeros(1, n);
• g(1, 1:n-k+1) = gx;
• G_MATRIX(1, :) = g;
• for i = 2:k
    temp = g(1, 1:n-1);
    g = [g(n) temp];
    G_MATRIX(i, :) = g;
• end
• G_MATRIX
• disp('CODE DETAILS')
• CODELENGTH = n
• INFOLENGTH = k

STEP10:
• incidence_matrix = zeros(noofpoints, noofpoints);
• for i = 1:length(lines(:, 1))
    ref = lines(i, :);
    for j = 1:2^s+1
        incidence_matrix(i, ref(j)+1) = 1;
    end
• end
• H_MATRIX = incidence_matrix
• sum(sum(rem(GMATRIX*H_MATRIX', 2)))
• disp('CODE DETAILS')
• CODELENGTH = n

• INFOLENGTH = k

4.10 SAMPLE OUTPUT OF CODE ‘2DPG’

Enter S Degree Of The Subfield 2

s = 2

Dimension Of The Projective Space

m = 2

Degree Of GF = 6

Size Of GF = 64

Prim_Poly = 1 1 0 0 0 0 1

FeedbakTaps = 0 0 0 0 1 1

Galois Field =

0 0 0 0 0 0 1
1 0 0 0 0 1 0
2 0 0 0 1 0 0
3 0 0 1 0 0 0
4 0 1 0 0 0 0
5 1 0 0 0 0 0
6 0 0 0 0 1 1
7 0 0 0 1 1 0
8 0 0 1 1 0 0
9 0 1 1 0 0 0
10 1 1 0 0 0 0
11 1 0 0 0 1 1
12 0 0 0 1 0 1
13 0 0 1 0 1 0
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>26</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>27</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>36</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
No. Of PointsOnPG = 21

Points Of PG =

0 0 0 0 0 0 1
1 0 0 0 0 1 0
2 0 0 0 1 0 0
3 0 0 1 0 0 0
4 0 1 0 0 0 0
5 1 0 0 0 0 0
6 0 0 0 0 1 1
7 0 0 0 1 1 0
8 0 0 1 1 0 0
9 0 1 1 0 0 0
10 1 1 0 0 0 0
11 1 0 0 0 1 1
12 0 0 0 1 0 1
13 0 0 1 0 1 0
14 0 1 0 1 0 0
15 1 0 1 0 0 0
16 0 1 0 0 1 1
17 1 0 0 1 1 0
18 0 0 1 1 1 1
19 0 1 1 1 1 0
20 1 1 1 1 0 0
Line0 =

0 1 6 8 18

Lines Of Pg: Line Construction Technique Is Due To Singer

LinesOfPG (Refer Fig.4.3)=

0 1 6 8 18
1 2 7 9 19
2 3 8 10 20
3 4 9 11 0
4 5 10 12 1
5 6 11 13 2
6 7 12 14 3
7 8 13 15 4
8 9 14 16 5
9 10 15 17 6
10 11 16 18 7
11 12 17 19 8
12 13 18 20 9
13 14 19 0 10
14 15 20 1 11
15 16 0 2 12
16 17 1 3 13
17 18 2 4 14
18 19 3 5 15
19 20 4 6 16
20 0 5 7 17
Fig. 4.3 Lines in Projective Geometry

SubFieldElements =

0 0 0 0 0 0 0
1 0 0 0 0 0 1
2 1 1 1 0 1 1
3 1 1 1 0 1 0

Allowed Set Condition 1

AllowedSet =

0 3 6 9 12 15 18 21 24 27 30 33 36
39 42 45 48 51 54 57 60

jSet =

0 3
The Roots = 

\[
0 \ 3 \ 6 \ 9 \ 12 \ 18 \ 24 \ 33 \ 36 \ 48
\]

gx = 1 +a(1)x + a(2)x^2 + a(3)x^3 +...........a(n-k)x^{n-k}

gx =

\[
1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1
\]

k= The length of information vector is equal to ( n- 3^s+1)=11

n = 21

k = 11

G_MATRIX (11 x 21 Matrix) =

\[
\begin{array}{ccccccccccccccccc}
1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
\end{array}
\]

Code Details

Code length = 21

Infolength = 11

H_MATRIX (21 x 21 Matrix) =

\[
\begin{array}{cccccccccccccccccccc}
1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
The Encoder design for a binary information sequence $i$ of length $k$, the encoded sequence $c$ of length ‘n’ is obtained by the relation

$$ c = I \ast G_{\text{MATRIX}} $$

Note that $n = 2^{s+2s+1}$ and $k = n-(3s+1)$. $c$ can be computed using length $k$ or $n-k$ shift registers. The syndrome vector is given by the relation

$$ y = H_{\text{MATRIX}} \ast r^T $$

where $r$ is the received vector given by $r = c+e$, $e$ being the channel induced error vector. If $e =$ all-zero vector of length ‘n’ indicating that
there are no channel induced errors, then \( \mathbf{y} = \) all-zero vector of length ‘n’.

4.11 CONCLUSIONS

The implementation of procedure to generate points, lines, generator polynomial, generator matrix, parity check matrix, construction of codes in PG\((2, 2^n)\) is illustrated. In the next chapter, the decoding of received vector using majority logic decoding technique is discussed.