1. Basic concepts of linear codes

Monday, June 6 2005
Presenter: J. Walker.
Material: 1.1 - 1.4
For Lecture: 1, 4, 5, 8
Problem Set: 6, 10, 14, 17, 19.

Presenting: Wednesday, June 8. Section 1.6 Problem Set: 35, 40, 41, 43

Coding Theory - trying to transmit information across a channel that may change the information. Goal: Coming up with efficient ways to send code, while detecting and correcting errors.

Picture - Figure 1.1 from book, pg. 2
Message - Encoder - codeword - Channel (noise inserted) - received word - Decoder - estimate of code word - Unencoder - Estimate of message.

Think of the message as \( x \in \mathbb{F}_q^k \). This is translated into a codeword \( c \in \mathbb{F}_q^n \), where \( n \geq k \). Let \( C \) be the set of all possible codewords. Some error is introduced, \( e \in \mathbb{F}_q^n \), resulting in the received word \( y = c + e \in \mathbb{F}_q^n \). The decoder takes \( y \) and produces \( \hat{c} \in \mathbb{F}_q^n \in C \), where our hope is that \( \hat{c} = c \). Finally, the unencoder produces an estimate of the message, \( \hat{x} \in \mathbb{F}_q^k \). Because the encoder is injective, we usually skip the estimate of message/encoder part.

Example of a Channel: Binary Symmetric Channel
Binary implies that \( q = 2 \), symmetric implies that the error is same for either 0 or 1. We have 2 possible errors -

(1) A “1” is received as a “0” or

(2) A “0” is received as a “1”.

We denote the probability of error in any given bit as \( p \), where \( 0 \leq p < \frac{1}{2} \). Probabilities:

- \( 0 \to 1: p \)
- \( 0 \to 0: 1 - p \)
- \( 1 \to 1: 1-p \)
- \( 1 \to 0: p \)

Example: Repetition Code
 Transmit each symbol \( n \) times, then do a “majority vote” to decode. If \( n = 3 \),
$q = 2$, we have 2 messages $(0, 1)$ and 2 codewords $(000, 111)$. We can correct at most 1 error in the sent codeword. For example, we would decode 101 as 111 because 111 is the most probable codeword sent. However, the code can detect up to $n-1$ errors. The problem with this code is that it is too expensive - in order to correct one error, you must triple the message size.

1.1. Definition of a Code.

**Definition 1.1.1.** A code $C$ of length $n$ over the alphabet $A$ is a subset of $A^n$. Elements of $C$ are called codewords.

From now on, $A = \mathbb{F}_q$ for some prime power $q$.

**Definition 1.1.2.** The code $C \subset \mathbb{F}_q^n$ is called a linear code if $C$ is a subspace of $\mathbb{F}_q^n$. The dimension of $C$ is its dimension as a vector space over $\mathbb{F}_q$.

We like linear codes because there is a lot of structure, making them easy to study and easy to encode.

**Notation:** “$C$ is a $[n,k]_q$ code” or “$C$ is a $[n,k]$ code” (when $q$ is understood or irrelevant) means $C$ is a linear code over $\mathbb{F}_q$ of length $n$ and dimension $k$.

**Warning:** There is only one vector space (up to isomorphism) of dimension $k$. Vector space isomorphisms are too general for use in coding theory because they don’t preserve Hamming weight.

**Terminology:** If $q = 2$, we call $C$ a binary code. If $q = 3$, we call $C$ a ternary code.

**Example:** The $n$-fold repetition code is a $[n,1]_q$ code. Our example was a $[3,1]_2$ code.

1.2. Generator Matrices and Parity Check Matrices.

**Definition 1.2.1.** Let $C$ be an $[n,k]_q$ code. A generator matrix for $C$ is any $k \times n$ matrix $G$ with entries in $\mathbb{F}_q$ such that the rows of $G$ form a basis for $C$.

**Remark:** If $G$ is a generator matrix for $C$, then

$$C = \{xG | x \in \mathbb{F}_q^k\}$$

**Definition 1.2.2.** Let $C$ be an $[n,k]_q$ code. Any set of $k$ linearly independent columns of $C$ is called an information set for $C$. More rigorously, let $C$ be an $[n,k]_q$ code and $G$ a generator matrix for $C$. An information set for $C$ is a set of integers $\{i_1, \ldots, i_n\} \subset \{1, \ldots, n\}$ such that the corresponding columns of $G$ are linearly independent vectors in $\mathbb{F}_q^k$. (This is independent of the generator matrix)

**Example:** If $C$ is an $n$-fold repetition code over $\mathbb{F}_q$. Then $C$ is an $[n,1]_q$ code. Generator matrix:

$$[1, \ldots, 1]$$

Every column forms (by itself) is an information set for $C$. Hence, we have $n$ information sets for $C$. 

Example: Let \( C \) be the code with the generator matrix

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Then, \( C \) is an \([7, 4]_2\) code. Information sets:

- \{1, 2, 3, 4\}
- \{1, 2, 3, 5\}
- \{1, 3, 4, 5\}
- \{1, 2, 4, 5\}
- \{1, 2, 3, 6\}
- \{1, 2, 3, 7\}

Redundancy Sets:

- \{2, 3, 4, 5\}
- \{4, 5, 6, 7\}

Definition 1.2.3. A generator matrix of the form

\[
[I_k | A]
\]

is said to be in standard form.

Not every code has a generator matrix \( G \) with standard form. For example, consider the code generated by the following matrix:

\[
\begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

In such case, \{1, \ldots, k\} is always an information set.

Definition 1.2.4. Let \( C \) be an \([n, k]_q\) code. A parity check matrix for \( C \) is an \((n - k) \times n\) matrix \( H \) over \( F_q \) such that

\[
C = \{c \in F_q^n : Hc^T = 0\}
\]

Note: Let \( T : F_q^k \to F_q^n \) be given by

\[
T(x) = xG
\]

where \( G \) is a generator matrix. Then, \( T \) is injective and the image of \( T \) is \( C \subset F_q^n \).

Further,

\[
F_q^n / C \cong F_q^{n-k}
\]

where \( \cong \) is a vector space isomorphisms. We get a short exact sequence:

\[
0 \to F_q^k \to F_q^n \to F_q^{n-k} \to 0
\]

(the kernel of one map is the image of the previous map). Label the map from \( F_q^n \to F_q^{n-k} \) as \( U \). Because it is a linear transformation between the fields, it is given as multiplication by some matrix \( M \) (size \( n \times (n - k) \)), so \( U(y) = yM \). Hence, \( H = M^T \).
Proposition 1.2.5. If $G = [I_k | A]$ is a generator matrix for $C$, then 
$$H = [-A^T | I_{n-k}]$$
is a parity check matrix for $C$.

Proof. Note that $H$ has size $(n-k) \times n$, so we just need to show that 
$$C = \{ c \in \mathbb{F}_q^k : Hc^T = 0 \}$$

We know already that 
$$C = \{ x \in G : x \in \mathbb{F}_q^k \}$$

So we have, 
$$c \in C \iff c = xG \text{ for some } x \in \mathbb{F}_q^k$$

Pick $c \in C$. Then, 
$$Hc^T = H(xG)^T = HG^T x^T$$
$$= (-A^T + A^T)x = 0$$

So, $C \subset \{ c \in \mathbb{F}_q^k : Hc^T = 0 \}$. By a dimension argument, as $\dim_{\mathbb{F}_q} C = k$ and since $H$ has rank $n-k$, 
$$\dim_{\mathbb{F}_q} \{ c \in \mathbb{F}_q^k : Hc^T = 0 \} = n - (n-k) = k$$

So the spaces are equal. \qed

Example: $n$-fold repetition code:

$$G = [1, \ldots, 1]_{1 \times n}$$
$$H = \begin{bmatrix} -1 \\ \vdots \\ I_{n-1} \\ -1 \end{bmatrix}$$

Example:

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$
$$H = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

1.3. Dual Codes.

Definition 1.3.1. Let $C$ be a code of length $n$ over $\mathbb{F}_q$. The dual code of $C$ is 
$$C^\perp := \{ y \in \mathbb{F}_q^n : x \cdot y = 0, \forall x \in C \}$$

where if $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_n)$ are in $\mathbb{F}_q^n$, then 
$$x \cdot y = \sum_{i=1}^n x_i y_i \in \mathbb{F}_q.$$ 

Proposition 1.3.2. If $C$ is an $[n, k]_q$ code with generator matrix $G$ and parity check matrix $H$, then $C^\perp$ is an $[n, n-k]_q$ code with generator matrix $H$ and parity check matrix $G$. 

Proof. Clearly, $C^\perp$ has length $n$. Then,
\[
y \in C^\perp \iff c \cdot y = 0, \forall c \in C
\]
\[
\iff (xG) \cdot y = 0, \forall x \in \mathbb{F}_q^k
\]
\[
\iff xGy^T = 0, \forall x \in \mathbb{F}_q^k
\]
\[
\iff Gy^T = 0 \quad (\text{since} \; rk(G) = k).
\]
So,
\[
C^\perp = \{ y \in \mathbb{F}_q^n : Gy^T = 0 \}
\]
i.e., $G$ is a parity check matrix for $C^\perp$.

Since $Hc^T = 0$ for all $c \in C$, every row of $H$ lives in $C^\perp$. Since $H$ has $n - k$ linearly independent rows and $\dim C^\perp = n - rk(G) = n - k$, this means that $H$ is a generator matrix for $C^\perp$.

\[\Box\]

**Definition 1.3.3.** If $C \subset C^\perp$, we say that $C$ is **self-orthogonal**. If $C = C^\perp$, we say $C$ is **self-dual**.

This is with respect to the standard inner product. One can also define self-dual, etc, with respect with other products.

Problem set: 6, 10, 14, 17, 19

1.4. Distances and Weights.

**Definition 1.4.1.** Let $x, y \in \mathbb{F}_q^n$. The **Hamming distance** from $x$ to $y$ is
\[
d(x, y) := \# \{ i : x_i \neq y_i \}
\]
The **Hamming weight** of $x$ is
\[
wt(x) = \# \{ i : x_i \neq 0 \} = d(x, 0)
\]

**Remark:** The Hamming distance make $\mathbb{F}_q^n$ into a **metric space**:
(1) $d(x, y) \geq 0$, and equality holds if and only if $x = y$.
(2) $d(x, y) = d(y, x)$
(3) $d(x, z) \leq d(x, y) + d(y, z)$, $\forall x, y, z \in \mathbb{F}_q^n$

**Definition 1.4.2.** Let $C \subset \mathbb{F}_q^n$ be a code. The **minimum distance** of $C$ is
\[
d = d_{\min}(C) = \min \{ d(x, y) : x, y \in C, \; x \neq y \}
\]

**Notation:** If $C$ is $[n, k]_q$ and has minimum distance $d$, we write $C$ is an $[n, k, d]_q$ code.

**Example:** $n$-fold repetition. Minimum distance - $n$. Hence, we have an $[n, 1, n]$ code. We can show the other example (Hamming code) is $[7, 4, 3]_2$.

**Facts:**
(1) If $C$ is linear, $d_{\min}(C) = w_{\min}(C)$ (exercise)
(2) If $C$ has minimum distance $d$, then $C$ can correct up to $\left\lfloor \frac{d - 1}{2} \right\rfloor$ errors.
Theorem 1.4.3. The following are equivalent:

1. The minimum distance, $d_{\text{min}}(C)$, is $d$
2. $H$ has a set of $d$ linearly dependent columns and no set of $d - 1$ linearly dependent columns
3. $d$ is the largest number such that every set of $n - d + 1$ positions contains an info set

Proof. Not that $Hx^T$ is a linear combination of columns of $H$; to say $x \in C$ has weight $d$ says $x$ gives a linear dependence among the columns of $H$. This proves that (1) $\iff$ (2).

Let $X$ be a set of $s$ coordinate positions of $\mathbb{F}_q^n$. Assume $X$ does not contain an info set. Let $G$ be any generator matrix for $C$ and let $B$ be the $k \times s$ matrix whose columns are those of $G$ indexed by the elements of $X$. As $X$ doesn’t contain an information set, the rank of $B$ is strictly less than $k$. Thus, there exists a nontrivial linear combination of rows of $B$ which is 0. Using the same linear combination, except on the rows of $G$, we get a nonzero codeword of weight at most $n - s$. So, $d_{\text{min}} \leq n - s$. Therefore, any set of size $\geq n - d + 1$ has an info set. We need that there exists a set of size $n - d$ with no information set. Take $X$ to be the columns on which a minimum weight code word is zero - this will be a set which contains no information set. (If $i_1, \ldots, i_k$, then the only code work $x = (x_1, \ldots, x_n)$ with $x_{i_1} = \cdots = x_{i_k} = 0$ is the 0 code word. Exercise). \qed

Corollary 1.4.4. For any $[n, k, d]$ code, we have $d \leq n - k + 1$.

This corollary is called the singleton bound.

Definition 1.4.5. Let $C \subset \mathbb{F}_q^n$. Define

$$A_i = \#\{x \in C : \text{wt}(x) = i\}.$$  

for $0 \leq i \leq n$. The sequence $A_0, \ldots, A_n$ is called the weight distribution of $C$.

Easy observations for $C$ linear:

- $A_0 = 1$
- If $d = d_{\text{min}}(C)$, then for $1 \leq i \leq d - 1$, $A_i = 0$. Further, $A_d \geq 1$.
- $\sum_{i=0}^{n} A_i = |C|$ ($= q^k$ for $C$ linear).

Theorem 1.4.6. Let $C$ be an $[n, k, d]$ self-orthogonal code. Then,

1. $1 = 1 = (1, \ldots, 1) \in C^\perp$
2. $k \leq \lfloor \frac{n}{2} \rfloor$
3. $d$ is even
4. $A_i = 0$ if $i$ is odd.
Set \( C_0 = \{ x \in C : \text{wt}(x) \equiv 0(\text{mod}4) \} \).

Then \( C_0 \) is a linear subcode of \( C \) of dimension at least \( k - 1 \).

**Proof.**

(1) Let \( c \in C \). By (4), the \( \text{wt}(c) \) is even. Hence, 
\[
1 \cdot c = \sum c_i \equiv 0(\text{mod}2)
\]
This implies that \( 1 \in C^\perp \).

(2) \( C \subset C^\perp \) so \( \text{dim} C \leq \text{dim} C^\perp \). Then, 
\[
k \leq \left\lfloor \frac{n}{2} \right\rfloor
\]
because \( k \in \mathbb{Z} \).

(3) Follows from (4)

(4) Let \( c \in C \) have weight \( i \). Then 
\[
0 = c \cdot c = \sum_{j=1}^{n} c_j^2 \equiv \text{wt}(c)(\text{mod}2)
\]
Note that \( c \cdot c = 0 \) because \( C \subset C^\perp \).

(5) Define \( f : C \rightarrow \mathbb{F}_2 \) by 
\[
f(c) = \begin{cases} 
0, & \text{if } \text{wt}(c) \equiv 0(\text{mod}4) \\
1, & \text{wt}(c) \equiv 2(\text{mod}4) 
\end{cases}
\]

Claim: \( f \) is a vector space homomorphism.

**Proof.** Let \( x, y \in \mathbb{F}_2^n \). Then 
\[
\text{wt}(x + y) = \text{wt}(x) + \text{wt}(y) - 2\# \{ i : x_i = y_i = 1 \}
\]
\[
= \text{wt}(x) + \text{wt}(y) - 2x \cdot y
\]
So for \( x, y \in C \), 
\[
\text{wt}(x + y) \equiv \text{wt}(x) + \text{wt}(y)(\text{mod}4).
\]
Hence, \( f \) is a vector space homomorphism. \( \square \)

Note that \( C_0 = \ker f \). If \( f \) is onto, then \( C/C_0 \cong \mathbb{F}_2 \) as a vector space (first isomorphism theorem), which implies that \( \text{dim} C_0 = k - 1 \).
If \( f \) is not onto, \( C_0 = \ker f = C \) so \( \text{dim} C_0 = k \). \( \square \)

**Generalizations**

(1) If \( C \) is \([n, k]_2\), set 
\[
C_e := \{ c \in C : \text{wt}(c) \text{ is even} \}
\]
Then \( C = C_e \) or \( C_e \) has \( \text{dim} k - 1 \).

(2) If \( C \) is \([n, k]_q\), set 
\[
C_e = \{ c \in C : \sum_{i=1}^{n} c_i = 0 \text{ in } \mathbb{F}_q \}.
\]
Elements of \( C_e \) are called “even-like” codewords. All others are called “odd-like.” Then \( C_e = C \) or \( C_e \) has \( \text{dim} k - 1 \).
Exercise #19 from the book.

\[ G = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 \\
\end{bmatrix} \]

is the generator matrix for \( H_3 = [7,4,2] \). By listing out elements in the code, we can determine that the weight distribution is \( A_0 = A_7 = 1 \), \( A_3 = A_4 = 7 \). Hexacode:

\[ G_6 = \begin{bmatrix}
1 & \omega & \omega \\
I_3 & \omega & 1 & \omega \\
\omega & \omega & 1 \\
\end{bmatrix} \]

over \( \mathbb{F}_4 \). \( \mathbb{F}_4 = \mathbb{F}_2(\omega) \) where \( \omega^2 + \omega + 1 = 0 \). Fact: every codeword in the hexacode has even weight.

• \( A_0 = 1 \)
• \( A_1 = 0 \)
• Any word of wt2 is a sum of multiples of 2 rows. Suppose \( \alpha x_i + \beta x_j \) has weight 2. Then, we’d need \( \alpha \) and \( \beta \) to solve the following equations:
  \[ \alpha + \beta \omega = 0 \]
  \[ (\alpha + \beta)\omega = 0 \]
  \[ \alpha \omega + \beta = 0 \]

Hence, from the second equation, \( \alpha = \beta \). From either one or three, we can instantly see that this is not possible. Hence \( A_2 = 0 \).

• \( A_3 = 0 \)
• \( A_4 = 4^3 - 1 - A_6 \)
• \( A_5 = 0 \)
• We want words of weight 6, so we want something of the form
  \[ \alpha x_1 + \beta x_2 + \gamma x_3 \]

Let’s say we count the number of words of weight 6 with \( \alpha = 1 \). Then since wt(\( y \)) = 6 \( \Leftrightarrow \) wt(\( \omega y \)) = 6 \( \Leftrightarrow \) wt(\( \omega^2 y \)) = 6, we multiply by 3 to get total number of code words of weight 6.

1.5. New Codes from Old. In this section, we will learn about 5 new code constructions.

Direct Sum Method
Let \( C_i \) be a \([n_i, k_i, d_i]_q\) code for \( i = 1,2 \). Define

\[ C = C_1 \oplus C_2 = \{ (c_1, c_2) : c_1 \in C_1, c_2 \in C_2 \} \]

Example: 1.5.8 in book. Let

\[ D = \{00, 11\} \]

which is a \([2,1,2]_2\) code with generator matrix \([1,1]\) and a parity check matrix \([1,1]\). Then,

\[ D \oplus D = \{0000, 0011, 1100, 1111\} \]

is a \([4,2,2]_2\) code with generator matrix

\[ \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\end{bmatrix} \]
and parity check matrix
\[
\begin{bmatrix}
  0 & 0 & 1 & 1 \\
  1 & 1 & 0 & 0 \\
\end{bmatrix}.
\]

**Theorem 1.5.1.** Let \( C_1 \) be a \([n_1, k_1, d_1]\) code and \( C_2 \) be a \([n_2, k_2, d_2]\) code. Then \( C_1 \oplus C_2 \) is an \([n_1+n_2, k_1+k_2, \min\{d_1, d_2\}]\) code with generator matrix
\[
\begin{bmatrix}
  G_1 & 0 \\
  0 & G_2 \\
\end{bmatrix}
\]
and parity check matrix
\[
\begin{bmatrix}
  H_1 & 0 \\
  0 & H_2 \\
\end{bmatrix}.
\]

**The \((u : u + v)\) construction**

Let \( C_i \) be an \([n_i, k_i, d_i]_q\) code for \( i = 1, 2 \). Then, define a new code as follows:
\[
C = \{ (u, u + v) : u \in C_1, v \in C_2 \}.
\]

If \( C_i \) has generator matrix \( G_i \) and parity check matrix \( H_i \) for \( i = 1, 2 \), then \( C \) has the following generator and parity check matrices:
\[
G = \begin{bmatrix}
  G_1 & G_1 \\
  0 & G_2 \\
\end{bmatrix}
\]
\[
H = \begin{bmatrix}
  H_1 & 0 \\
  -H_1 & H_2 \\
\end{bmatrix}
\]

\( C \) is a \([2n, k_1 + k_2, \min\{2d_1, d_2\}]\) linear code.

**Example:** Let \( C_1 \) be a \([4, 3, 2]_2\) code with
\[
G_1 = \begin{bmatrix}
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 1 \\
\end{bmatrix}
\]
and \( C_2 \) a \([4, 1, 4]_2\) code with
\[
G_2 = [1, 1, 1, 1]
\]

Then, we can write out \( C_1 \):
\[
C_1 = \{0000, 1000, 0101, 0011, 1101, 1011, 0110, 1110\}
\]
\[
C_2 = \{0000, 1111\}
\]
\[
C = \{00000000, 00001111, 10010000, 10001111, \ldots\}
\]

Note that \( C \) is a \([8, 4, 4]_2\) code with
\[
G = \begin{bmatrix}
  1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
  0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
  0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\end{bmatrix}
\]

**Extending Codes**

Let \( C \) be an \([n, k, d]_q\) code. Then, define
\[
\hat{C} = \{ x_1, \ldots, x_{n+1} \in \mathbb{F}_q^{n+1} : x_1 + \cdots + x_{n+1} = 0 \}
Our parity check matrix has the following form:

\[ \hat{H} = \begin{bmatrix}
    1 & \ldots & 1 & 1 \\
    0 \\
    H \\
    \vdots \\
    0
\end{bmatrix} \]

This is our in-class exercise.

**Example** 1.5.4 in the book. Let \( \mathcal{H}_{3,2} \) be a \([4, 2, 3]_3\) code with

\[
G = \begin{bmatrix}
    1 & 0 & 1 & 1 \\
    0 & 1 & 1 & -1
\end{bmatrix}
\]

\[
H = \begin{bmatrix}
    -1 & -1 & 1 & 0 \\
    -1 & 1 & 0 & 1
\end{bmatrix}
\]

\[
\mathcal{H}_{3,2} = \{(0, 0, 0, 0), (1, 0, 1, 1), (-1, 0, -1, -1), (0, 1, 1, -1), \\
(0, -1, -1, 1), (1, 1, -1, 0), (1, -1, 0, -1), (-1, 1, 0, 1)\}
\]

\[
\hat{\mathcal{H}}_{3,2} = \{0000, 10110, -10 - 10 - 10, 011 - 1 - 1, 0 - 1 - 111, -1 - 110\}
\]

\[
\hat{G} = \begin{bmatrix}
    1 & 0 & 1 & 1 & 0 \\
    0 & 1 & 1 & -1 & -1
\end{bmatrix}
\]

\[
\hat{H} = \begin{bmatrix}
    1 & 1 & 1 & 1 & 1 \\
    -1 & -1 & 1 & 0 & 0 \\
    -1 & 1 & 0 & 1 & 0
\end{bmatrix}
\]

So, \( d = 3 \).

From the book,

\[
\hat{d} = \begin{cases} 
    d, & d = d_e \\
    d + 1, & d = d_o < d_e
\end{cases}
\]

**Puncturing Codes**

Let \( C \) be an \([n, k, d]_q\) code. Let \( T \) be a set of positions of size \( t \).

For \( t = 1 \), say \( T = \{i\} \), then

\[
C^T = \{x_1, \ldots, \hat{x}_i, \ldots, x_n : x_1, \ldots, x_n \in C\},
\]

so \( C^T \) is a \([n - t, k^*, d^*] \) code (where \( k^* \) and \( d^* \) may not necessarily be \( k \) and \( d \), respectively).

**Example**: 1.55.2 in the book.

Let \( C \) be a \([5, 2, 2]_2\) code with

\[
G = \begin{bmatrix}
    1 & 1 & 0 & 0 & 0 \\
    0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

Let \( T_1 = \{1\}, T_5 = \{5\} \).

\[
C = \{00000, 11000, 00111, 11111\}
\]

\[
C^{T_1} = \{0000, 1000, 0111, 1111\}
\]

\[
C^{T_5} = \{0000, 1100, 0011, 1111\}
\]
Theorem 1.5.2. Assume that |T| = 1. For a more general version, we can easily induct on the size of T.

(i) If d > 1, $C^T$ is an $[n-1, k, d^*]$ where $d^* = d-1$ if C has a minimum weight code word which is nonzero in the $i^{th}$ coordinate, $d^* = d$ otherwise.

(ii) If d = 1, then $C^T$ is an $[n-1, k, 1]$ code if C has no code word of weight 1 whose nonzero entry is in coordinate i, otherwise, if $k > 1$, $C^T$ is an $[n-1, k-1, d^*]$ code with $d^* \geq 1$.

Example:

$C = \{000, 001, 110, 111\}$ over $\mathbb{F}_2$. $C$ is a $[3, 2, 1]_2$ code with generator matrix

$$G = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Dropping the first coordinate,

$$G_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

So, $C_{T1}$ is a $[2, 2, 1]_2$ code. Dropping the third coordinate,

$$G_3 = [1, 1]$$

$C_{T3}$ is a $[2, 1, 2]_2$ code.

Shortening Codes
Assume $C$ is a $[n, k, d]_q$ code and $T$ is a set of $t$ positions. Define

$$C(T) := \{c \in C : c_i = 0, \forall i \in T\}$$

Then, the shortened code $C_T$ is defined as

$$C_T := C(T)^T$$

Example:

Consider a code $C$ which is $[6, 3, 2]_2$ and

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Let $T = 5, 6$.

$C = \{000000, 100111, 010111, 001111, 110000, 101000, 011000, 111111\}$

$C_T = \{0000, 1100, 1010, 0110\}$

Theorem 1.5.3. Let C be a $[n, k, d]_q$ code and $T$ be a set of $t$ positions.

(i) $(C^\perp)_T = (C^T)^\perp$ and $(C^\perp)^T = (C_T)^\perp$.

(ii) If $t < d$, then $C^T$ and $C^\perp$ have dimensions $k$ and $n - t - k$, respectively.

(iii) If $t = d$ and $T$ is the set of coordinates where a minimum weight codeword is nonzero, then $C^T$ and $(C^\perp)_T$ have dimension $k - 1$ and $n - d - k + 1$ respectively.
Exercise 27:

(a) $C = H_{3,2}$ is the $[4, 2, 3]_3$ code with

\[
G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & -1 \end{bmatrix}.
\]

Let $C_1$ be the code punctured on the right and then extended on the right.

\[
G_1 = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}
\]

\[
H_1 = \begin{bmatrix} -1 & -1 & 1 & 0 \\ -1 & -1 & 0 & 1 \end{bmatrix}
\]

From this, we can tell that $d = 2$.

(b) $C$ an $[n, k, d]_q$ code. Let $C_1$ be punctured, then extended. Prove that $C = C_1$ if and only if $C$ is even-like.

For the forward direction, suppose $C = C_1$. As $C_1$ is even-like, then we get instantly that $C$ is even-like.

For the backwards direction, suppose that $C$ is even-like. Then, for all $x \in C$,

\[
x_1 + \cdots + x_n = 0.
\]

Therefore,

\[
x_n = -(x_1 + \cdots + x_{n-1}).
\]

Also, for $x' = (x_1, \ldots, x_{n-1}, y)$ where

\[
x_1 + \cdots + x_{n-1} + y = 0
\]

So we get

\[
y = -(x_1 + \cdots + x_{n-1}) = x_n,
\]

\[
x = x' \in C_1
\]

$C \subset C_1$.

On the other hand, suppose that $z' \in C_1$ and $z' = z_1, \ldots, z_{n-1}, q$. Then, there exists a $z \in C$ with $z = z_1, \ldots, z_n$. Hence, $z' = z \in C$ and $C_1 \subset C$. Thus, $C_1 = C$.

(c) Suppose that $C \subset C^\perp$ and $1 \in C$. Then,

\[
x \cdot y = 0, \forall x, y \in C.
\]

\[
x \cdot 1 = 0, \forall x \in C
\]

Thus,

\[
0 = x \cdot 1 = x_1 + \cdots + x_n
\]

for all $x \in C$, and $C$ is even-like. By (b), $C = C_1$. 

(d) Prove $C = C_1$ if and only if $1 \in C^\perp$.
For the backwards direction, assume that $1 \in C^\perp$. Then,
$$
x \cdot 1 = 0, \forall x \in C
$$
$$
0 = x \cdot 1 = x_1 + x_2 + \cdots + x_n
$$
Thus, $C$ is even-like and $C = C_1$.
For the forwards direction, $C = C_1$ so $C$ is even-like and $x_1 + \cdots + x_n = 0$.
Therefore, for all $x \in C$,
$$
x \cdot 1 = 0
$$
$$
1 \in C^\perp
$$

**Exercise 29:**
Let $C$ be generated by
$$
G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}
$$
Let $T_1 = \{1, 2\}$ and $T_2 = \{1, 3\}$. Show that $(C^\perp)_{T_i} = (C_{T_i})^\perp$ for $i = 1, 2$.

**Exercise 32:**
Prove that the generator and parity check matrices for the code $(u|u + v)$ are
$$
G = \begin{bmatrix} G_1 & G_1 \\
0 & G_2 \end{bmatrix}
$$
and
$$
H = \begin{bmatrix} H_1 & 0 \\
-H_1 & H_2 \end{bmatrix},
$$
respectively.

**Exercise 33:**
Prove that $(u|u + v)$ construction using $[n, k_i, d_i]$ codes $C_i$ produce dimensions $k = k_1 + k_2$ and $d_{\text{min}} = \min\{2d_1, d_2\}$.

1.6. **Permutation Equivalent Codes.** In this section, we aim to answer the question “when are two codes essentially the same?”

As has been noted before, vector space isomorphisms won’t work as they don’t preserve weight, which is an essential property of each code.

**Definition 1.6.1.** Two linear codes $C_1$ and $C_2$ are called **permutation equivalent** provided there is a permutation of coordinates which sends $C_1$ to $C_2$.

**Remark:** It is often convenient to express the permutation between codes in a permutation matrix, which is a matrix with exactly one 1 in each row and column and zeros elsewhere. Another way to see that two codes are permutation equivalents is if there is a permutation matrix such that $G_1$ is a generator for $C_1$ if and only if $G_1P$ is a generator for $C_2$. We define:
$$
CP = \{y : y = xP \text{ for some } x \in C_1\}$$
Facts about permutation matrices:
- If $P$ and $Q$ are $n \times n$ permutation matrices, then so is $PQ$ (matrix multiplication).
- $I_n$ is a permutation matrix
- If $P$ is a permutation matrix, there exists an inverse permutation matrix, $P^{-1}$.

Exercise: 35 from the book.
Suppose that $C_1$ and $C_2$ are permutation equivalent codes where $C_1P = C_2$ for some permutation matrix $P$. Prove that:
(a) $C_1^\perp P = C_2^\perp$, and
(b) if $C_1$ is self-dual, so is $C_2$.

Example: 1.6.1 from the book.
Let $C_1$, $C_2$, and $C_3$ be binary codes with generator matrices

\[
G_1 = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
G_2 = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 1
\end{bmatrix}
\]

\[
G_3 = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Note that $C_1$ and $C_2$ are permutation equivalent with permutation matrix:

\[
P = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0
\end{bmatrix}
\]

We can also tell that $C_1$ and $C_3$ are not permutation equivalent - $C_1$ is a self-dual code while $C_3$ is not.

Example: Let $C_1$ and $C_2$ have generator matrices

\[
G_1 = \begin{bmatrix}
1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 1
\end{bmatrix}
\]

\[
G_2 = \begin{bmatrix}
1 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

respectively. Note that there is not going to be a permutation matrix $P$ so that $G_1P = G_2$. 

However, when we write out the codes:

\[
C_1 = \{00000, 11000, 01010, 00101, 10010, 11101, 01111, 10111\}
\]

\[
C_2 = \{00000, 10001, 00011, 01100, 10010, 11101, 01111, 11110\}
\]

we notice that the two codes are in fact permutation equivalent.

**Theorem 1.6.2.** Let \( C \) be a linear code

(i) \( C \) is permutation equivalent to a code which has generator matrix in standard form

(ii) If \( I \) and \( R \) are information and redundancy positions, respectively, for \( C \), then \( R \) and \( I \) are information and redundancy positions, respectively, for the dual code \( C^\perp \).

**Note:** Part (ii) states that information and redundancy positions switch for the dual code

**Proof:** For (i), apply elementary row operations to any generator matrix of \( C \). This will produce a new generator matrix of \( C \) which has columns the same as those in \( I_k \), but possibly in a different order. Now choose a permutation of the columns of the new generator matrix so that these columns are moved to the order that produces \([I_k|A]\). The code generated by \([I_k|A]\) is equivalent to \( C \).

If \( I \) is an information set for \( C \), then by row reducing a generator matrix for \( C \), we obtain columns in the information positions that are the columns of \( I_k \) in some order. As above, choose a permutation matrix \( P \) to move the columns so that \( CP \) has generator matrix \([I_k|A]\); \( P \) has moved \( I \) to the first \( k \) coordinate positions. By Theorem 1.2.1, \((CP)^\perp \) has the last \( n-k \) coordinates as information positions. By Exercise 35, \((CP)^\perp = C^\perp P \), implying that \( R \) is a set of information positions for \( C^\perp \), proving (ii). \( \square \)

Instead of thinking of the permutation as a matrix that we multiply a code \( C \) by, think of the permutation as an operation that we apply to \( C \). This way, we can write the permutations much more succinctly in cycle form. To do this, let \( \sigma \in Sym_n \) and \( x = (x_1, x_2, \ldots, x_n) \in C \). Then,

\[
x \sigma = (x_{1\sigma^{-1}}, x_{2\sigma^{-1}}, \ldots, x_{n\sigma^{-1}})
\]

**Example:** 1.6.3 from the book.

Let \( n = 3 \), \( x = (x_1, x_2, x_3) \), and \( \sigma = (1, 2, 3) \). Then,

\[
x \sigma = (x_3, x_1, x_3)
\]

We can also write out the permutation matrix from \( \sigma \):

\[
P = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
\]

**Note:** If \( \sigma, \tau \in Sym_n \), then \( x(\sigma \tau) = (x \sigma) \tau \).

Of course, a coordinate permutation need not map a code \( C \) to a new code.
**Definition 1.6.3.** If a permutation maps $C$ to itself, then it is called an *(permutation) automorphism*. The set of all permutation automorphisms of a group forms the *permutation automorphism group* of $C$.

**Remark:** In this book, we denote the permutation automorphism group as $P Aut(C)$. If a code $C$ is of length $n$, then $P Aut(C)$ is a subgroup of the symmetric group $Sym_n$.

**Example:** Let $C$ be the $[n, 1]$ binary repetition code. Then, $P Aut(C) = Sym_n$ because all permutations are automorphisms.

**Example:** Consider the $[7, 4]_2$ binary code $H_3$, with the following generator matrix:

$$G = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Show that the following are automorphisms:

1. $\sigma_1 = (1, 2)(5, 6)$:

   $$G\sigma_1 = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}$$

   $G\sigma_1$ is simply the matrix $G$ with the rows 1 and 2 switched, and thus still generates $C$.

2. $\sigma_2 = (1, 2, 3)(5, 6, 7)$

   $$G\sigma_2 = \begin{bmatrix}
0 & 1 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}$$

   Again, $G\sigma_2$ is simply $G$ with the rows 1, 2, and 3 permuted. It still provides a basis for $C$.

3. $\sigma_3 = (1, 2, 4, 5, 7, 3, 6)$

   $$G\sigma_3 = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}$$

   Through simple row operations, we can transform $G\sigma_3$ into $G$, so $G\sigma_3$ provides a basis for $C$.

These three permutations generate a simple, non-abelian group of order 168, which turns out to be a very special group, $PSL(2, 7)$.

**Exercise:** How many unique groups are there that are permutation equivalent to $H_3$?

The original solution was fixed (aka, completely rewritten) by Judy Walker to reflect the fact that, well, it was incorrect.
From Algebra, we know the old adage “The length of the orbit is the index of the stabilizer.” In this case, we have the symmetric group on 7 symbols acting on the set of subsets of $\mathbb{F}_2^7$ by permuting the coordinates of $\mathbb{F}_2^7$. The “orbit” of the code is the set of all codes equivalent to the code. The “stabilizer” of the code is the set of all elements of the symmetric group which fix the code (as a set), i.e., the permutation automorphism group of the code. So since we know the stabilizer has order 168 and since we know the order of the symmetric group is 7!, we have that the length of the orbit is $7!/168 = 30$. In other words, the number of distinct codes equivalent to our original code (counting our original code) is 30.

Here are some useful properties to know about the permutation automorphism group:

**Theorem 1.6.4.** Let $C$, $C_1$, and $C_2$ be codes over $\mathbb{F}_q$. Then:

(i) $P\text{Aut}(C) = P\text{Aut}(C^\perp)$,

(ii) if $q = 4$, $P\text{Aut}(C) = P\text{Aut}(C^{\perp n})$, and

(iii) if $C_1 P = C_2$ for a permutation matrix $P$, then $P^{-1} P\text{Aut}(C_1) P = P\text{Aut}(C_2)$.

Proof is an exercise.

**Example:** 1.6.5 from the book.

Let $C$ be a binary code with the following generator matrix:

$$G = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

Let $C_1^*$ and $C_5^*$ be $C$ punctured on coordinates 1 and 5, respectively. Then, they are generated by the following matrices:

$$G_1^* = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

$$G_5^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$G_1^*$ generates all odd weight codewords, while $G_5^*$ generates all even weight codewords. Hence, they cannot be permutation equivalent.

**Definition 1.6.5.** The group $P\text{Aut}(C)$ is **transitive** as a permutation group if for every ordered pair $(i, j)$ of coordinates, there is a permutation in $P\text{Aut}(C)$ which sends coordinate $i$ to coordinate $j$.

If we know that $P\text{Aut}(C)$ is transitive, then we cannot have the case of the previous example.

(This lemma is used to prove part (i) of the next theorem, and is not in the book.)

**Lemma 1.6.6.** If $C$ is a code and $P$ a permutation where $jP = i$, then

$$(CP)_i^* = C_j^* P_j^*$$

where $P_j^*$ is the permutation punctured at $j$ (remove column $i$ and row $j$).

**Theorem 1.6.7.** Let $C$ be an $[n, k, d]$ code.

(i) Suppose that $P\text{Aut}(C)$ is transitive. Then the $n$ codes obtained from $C$ by puncturing $C$ on a coordinate are permutation equivalent.
Suppose that $\text{PAut}(\hat{C})$ is transitive. Then the minimum weight $d$ of $C$ is its minimum odd-like weight, $d_o$. Furthermore, every minimum weight codeword of $C$ is odd-like.

**Proof.**

(i) Left to an exercise (42). Suppose that $\text{PAut}(C)$ is transitive. Fix $1 \leq i, j \leq n$, and consider $C^*_i$ and $C^*_j$. Because $\text{PAut}(C)$ is transitive, there is a $P \in \text{PAut}(C)$ that sends $i$ to $j$. Because $P \in \text{PAut}(C)$, $CP = C$.

Now, puncture $CP$ on the $j^{th}$ coordinate and $C$ on the $j^{th}$ coordinate. Because of the lemma, $C^*_j = (CP)^*_j = C^*_i P^*_i$. Hence, $C^*_j$ is permutation equivalent to $C^*_i$.

(ii) Again, assume that $\text{PAut}(\hat{C})$ is transitive. Applying (i) to $\hat{C}$ we conclude that puncturing $\hat{C}$ on any coordinate gives a code permutation equivalent to $C$. Let $c$ be a minimum weight vector of $C$ and assume that $c$ is even-like. Then $\text{wt}(\hat{c}) = d$, where $\hat{c} \in \hat{C}$ is the extended vector. Puncturing $\hat{C}$ on a coordinate where $c$ is nonzero gives a vector of weight $d - 1$ in a code permutation equivalent to $C$, a contradiction.

\[ \square \]

Thursday, June 9, 2005
Presenters: J. Brown Kramer, J. DeVries
Material: 1.8, 1.9, 1.10
For Lecture: 60
Problem Set: 59, 69, 62

### 1.7. Binary Hamming Codes.

Simplest example of an error correcting code

A binary hamming code $H_{2,r}$ has a parity check matrix $H_{2,r}$ consisting of all possible non-zero binary columns of length $r$. For example,

\[
H_{2,r} = \begin{bmatrix}
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 1
\end{bmatrix}
\]

The code for which $H_{2,r}$ is the parity check matrix is called the binary Hamming code of length $2^r - 1$. Its parameters are $[2^r - 1, 2^r - 1 - r, 3]$.

**Theorem 1.7.1.** Any $[2^r - 1, 2^r - r - 1, 3]_2$ code is permutation equivalent to the binary Hamming code of length $2^r - 1$.

**Exercise:** Any $[8, 4, 4]_2$ code is equivalent to $\hat{H}_3$.

**Other Fields:** $\mathbb{F}_q$

Construct a parity check matrix $H_{q,r}$ whose columns consist of exactly 1 non-zero vector from each 1-dimensional subspace of $\mathbb{F}_q^r$. The code $\mathcal{H}_{q,r}$ is the one for which $H_{q,r}$ is a parity check matrix. Parameters: $[(q^r - 1)/(q - 1), (q^r - 1)/(q - 1) - r, 3]_q$

**Theorem 1.7.2.** Any $[n_{q,r}, k_{q,r}, 3]_2$ code is monomially equivalent to the Hamming code $\mathcal{H}_{q,r}$.

Suppose $H_{q,r}$ is the generator matrix?

\[
\mathcal{H}_{2,3} = \{0000000, 0001111, 0110011, \ldots, 1101001\}
\]
All nonzero vectors of $\mathcal{H}_{q,r}^⊥$ have weight $q^r - 1$. The dual of a Hamming code is a simplex code ($[(q^r - 1)/(q - 1), r, q^r - 1]$).

1.8. Binary Golay Codes. Let $G_{24}$ be the $[24, 12]$ code with generator matrix $G_{24} = [I_{12} | A]$, where $A$ is defined as follows:

$$A = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1
1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}$$

Properties of $G_{24}$:

- Self-dual
- All codewords have length divisible by 4
- $[24, 12, 8]$ code
- Example of an almost-perfect code.

We can puncture $G_{24}$ anywhere to get $G_{23}$. This is a perfect code, and we call $G_{23}$ the binary Golay Code. We call $G_{24}$ the extended binary Golay Code.

**Ternary Colay Codes:**

$G_{12}$ is generated by $[I_6 | A]$ where

$$A = \begin{bmatrix}
0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1
1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1
1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1
1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1
1 & 1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1 & 1 & 0 & 1 & -1 & -1 & -1
\end{bmatrix}$$

$G_{12}$ is a $[12, 6, 6]$ self dual code. By puncturing on any coordinate, we get $G_{11}$ which is a perfect code.

1.9. Reed-Muller Codes. Here, we employ recursive construction techniques.

Let $m > 0$. Then, define

$$R(0, m) := [1, \ldots, 1]_{1,2^m}$$

$$R(m, m) := \mathbb{F}_2^{2^m}$$

For $0 < r < m$,

$$R(r, m) = \{ (u, u + v) : u \in R(r, m - 1), v \in R(r - 1, m - 1) \}$$

Note we can write down the generator matrix:

$$G = \begin{bmatrix}
G(r, m - 1) & G(r, m - 1) \\
0 & G(r - 1, m - 1)
\end{bmatrix}$$
Example:

\[
R(0, 1) = \{00, 11\}
\]
\[
R(1, 1) = \{00, 01, 11\}
\]
\[
R(0, 2) = \{0000, 1111\}
\]
\[
R(1, 2) = \begin{bmatrix}
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]
\[
R(1, 3) = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix}
\]

Theorem 1.9.1. Fix \( m > 0 \).

1. \( R(i, m) \subset R(j, m), 0 \leq i \leq j \leq m \)

2. \( \dim R(r, m) = \sum_{i=0}^{r} \binom{m}{i} \)

3. \( \min wt R(r, m) = 2^{m-r} \)

4. \( R(m, m) = \{0\} \)

\( R(r, m) \perp = R(m - r - 1, m), 0 \leq r < m \)

Proof. For 1, we induct on \( m \). Assume that \( R(i, m - 1) \subset R(j, m - 1) \) for \( 0 \leq i \leq j < m \). Assume \( 0 \leq k \leq l < m \):

\[
R(k, m) = \{(u, u + v) : u \in R(k, m - 1), v \in R(k - 1, m - 1)\}
\]
\[
\subset \{u, u + v) : u \in R(l, m - 1), v \in R(l - 1, m - 1)\} = R(l, m)
\]

For 2, we already know that

\[
\dim R(m, m) = 2^m = \sum_{i=0}^{m} \binom{m}{i}
\]

and

\[
\dim R(0, m) = 1 = \binom{m}{0}
\]

Now, for induction, assume that

\[
\dim R(i, m - 1) = \sum_{j=1}^{i} \binom{m-1}{j}
\]

Then,

\[
\dim R(r, m) = \dim R(r, m - 1) + \dim R(r - 1, m - 1)
\]
\[
= \sum_{i=0}^{r} \binom{m-1}{i} + \sum_{i=0}^{r-1} \binom{m-1}{i}
\]
\[
= \binom{m-1}{0} + \sum_{i=1}^{r} \binom{m-1}{i-1} + \binom{m-1}{i}
\]
\[
= \sum_{i=0}^{r} \binom{m}{i}
\]
For 3, note that $R(0, m)$ is repetition code of length $2^m$ and has min wt $2^m$. Then,
$$\min \text{wt} R(m, m) = 1$$

For induction, assume that
$$\min \text{wt} R(i, m - 1) = 2^{m - 1 - i}$$

Then,
$$\min \text{wt} R(i, rm) = \min(2^{m - 1 - r}, 2^{m - 1 - r + 1}) = 2^{m - i}$$

□

For duals, let $r = (m - 1)/2$. Then $R((m - 1)/2, m)$ is self dual.

1.10. Encoding, Decoding, and Shannon’s Theorem. We consider how we will decode the codeword $c$.

If $c \in \mathbb{F}_q^n$ and $y \in \mathbb{F}_q^n$, then
- $p(y|c)$ - probability that $y$ was sent, given $c$ was received.
- $p(c|y)$ - probability that $c$ was sent, given $y$ was received.

So, assuming that we have our BSC,

$$p(y_i|c_i) = \begin{cases} q, & y_i \neq c_i \\ 1 - q, & y_i = c_i \end{cases}$$

$$p(y|c) = \prod_{i=1}^n p(y_i|c_i) = q^{d(y,c)} (1 - q)^n - d(y,c)$$

$$p(y_i|c_i) = \left( \frac{q}{1 - q} \right)^{d(y,c)} (1 - q)^n$$

Because we can assume that $0 < q < \frac{1}{2}$, $0 < \frac{a}{1 - q} < 1$. Hence, $p(y|c)$ is maximized when $d(y, c)$ is minimized.

Minimizing $d(y, c)$ is called nearest neighbor decoding, while minimizing $p(y|c)$ is called maximum likelihood decoding. For the BSC, these two techniques are identical.

We know that $c + e = y$; so, we seek $e$ with the smallest weight so that $y - e = c$.

Definition 1.10.1. The sphere of radius $r$ about $u \in \mathbb{F}_q^n$ is
$$S_r(u) = \{ v \in \mathbb{F}_q^n : d(u, v) \leq r \}$$

Theorem 1.10.2. If $d$ is the minimum distance of $C$ and $t = \left[ \frac{d - 1}{2} \right]$, then spheres of radius $t$ about distinct codewords are disjoint.

Corollary 1.10.3. Let $d$ be the minimum distance of $C$ and $t = \left[ \frac{d - 1}{2} \right]$. If $c$ is sent and $y$ is received, and less than $t$ errors have occurred in transmission, then $c$ is the unique codeword closest to $y$. 
**Definition 1.10.4.** The packing radius of a code $C$ is the largest radius of spheres centered at codewords such that the spheres are pairwise disjoint.

**Theorem 1.10.5.** Let $C$ be an $[n, k, d]_q$ code. Then,

(i) packing radius of $C$ is precisely $t = \left\lfloor \frac{d - 1}{2} \right\rfloor$.

(ii) ML decoding always correctly decodes vectors with $t$ or fewer errors occurring. Further, ML decoding will not always do so if there are more than $t$ errors.

**Syndrome Decoding:**

Let $C$ be $[n, k, d]_q$. Notice that $C$ is an additive subgroup of $\mathbb{F}_q^n$, so the $n - k$ cosets $x + C$ ($x \in \mathbb{F}_q^n$), each having $q^k$ elements, partition $\mathbb{F}_q^n$ (and $x, y$ are in the same coset iff $y - x \in C$).

**Definition 1.10.6.** The weight of a coset is the weight of the smallest weight vector in the coset. Any vector in a coset of the smallest weight is a coset leader.

Note that in $0 + C$, $0$ is the only coset leader.

**Exercise:** Every coset of weight at most $t = \left\lfloor \frac{d - 1}{2} \right\rfloor$ has a unique coset leader.

**Definition 1.10.7.** Choose a parity check matrix $H$ for $C$. The syndrome of $x \in \mathbb{F}_q^n$ is $\text{syn}(x) = Hx^T$.

**Theorem 1.10.8.** Two vectors $x, y \in \mathbb{F}_q^n$ are in the same coset iff they have the same syndrome.

Therefore, we can define $C_s$ to be the coset of $C$ consisting of syndrome $s$ vectors.

Remember, in ML decoding, we receive $y$ and seek $e$ of minimum weight such that $y - e \in C$.

**Decoding Algorithm:**

1. Make a table of syndromes and coset leaders.
2. Receive $y$ and find $\text{syn}(y) = Hy^T := s$.
3. Find coset leader $e_s$ corresponding to syndrome $s$. Then, $y - e_s$ is the codeword $c$.

In order to make a table of syndromes, we will have $q^{n-k}$ entries.

**Erasures**

Up to this point, we assume that the error is a transmitted symbol is read as a different symbol. We now consider a second kind of error - erasures. An erasure is where a transmitted symbol is unreadable. We will know the location of erasures, but don’t know the locations of the errors.
Suppose $c \in C$ and $y$, with $\nu$ errors and $\epsilon$ errors. If $\epsilon > d$, then we cannot guarantee that $y$ can be corrected.

**Theorem 1.10.9.** Let $C$ be $[n, k, d]$. If a codeword $c$ is sent and $y$ is received, where $\nu$ errors and $\epsilon$ erasures have occurred, then $c$ is the unique codewords in $C$ closest to $y$ provided $2\nu + \epsilon < d$.

**Recall:** We have a BSC with crossover probability $q$ on which we do syndrome decoding.

**Definition 1.10.10.** The **word error rate** is the probability that the decoder makes an error, averaged over all codewords of $C$. We denote this as $P_{\text{err}}$.

When does decoder error occur? Define

$$\hat{c} = \arg \max_{c \in C} \text{prob}(y|c)$$

is not the originally transmitted word $c$ (assuming $y$ is received).

Our syndrome decoder is correct when $y - c$ is the chosen coset leader. Hence, the probability that the syndrome decoder is right is

$$\sum_{i=0}^{n} \alpha_i q^i (1-q)^{n-i}$$

where $\alpha_i$ is the number of cosets of weight $i$. Then,

$$P_{\text{err}} = 1 - \sum_{i=0}^{n} \alpha_i q^i (1-q)^{n-i}$$

**Example:** 1.11.9 from the book

Assume that we have $2^4 = 16$ binary messages. We would like to compare the trivial code (unencoded) to $\mathcal{H}_3$.

1. Unencoded - $[4,4]$ code over $\mathbb{F}_2^4$. Hence, the unique coset is the code itself.

Then,

$$P_{\text{err}} = 1 - q^0 (1-q)^4 = 1 - (1-q)^4$$

2. $\mathcal{H}_3$:

$$P_{\text{err}} = 1 - \sum_{i=0}^{n} \alpha_i q^i (1-q)^{n-i}$$

**Definition 1.10.11.** The **capacity** of a channel is

$$C(q) = 1 - q \log_2 q + (1-q) \log_2 (1-q)$$

**Definition 1.10.12.** If $C$ is a $[n, k]$ code, then $\frac{k}{n}$ is the **information rate**. This gives a measure of how much information is being transmitted.

**Theorem 1.10.13.** **Shannon’s Theorem:** Let $\delta > 0$ and $R < C(q)$. Then for large enough $n$, there exists an $[n, k]_2$ code $C$ with $\frac{k}{n} \geq R$ such that $P_{\text{err}} < \delta$, when $C$ is a code with our assumptions. Furthermore, no such code exists with $R > C(q)$.
1.11. Sphere Packing Bound. We start to look at bounds on the size of codes.

Definition 1.11.1. We define $B_q(n, d)$ to be the maximum number of code words in a linear code over $\mathbb{F}_q^n$ of length $n$ and minimum weight $d$. $A_q(n, d)$ is the maximum number of code words in any arbitrary code over $\mathbb{F}_q^n$ of length $n$ and minimum weight $d$.

**Theorem 1.11.2. Sphere Packing Bound**

$$B_q(n, d) \le A_q(n, d) \le \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i}$$

where $t = \lfloor \frac{d-1}{2} \rfloor$.

**Proof.** Let $C$ be a code over $\mathbb{F}_q$ (possibly nonlinear) of length $n$ and minimum distance $d$ such that $C$ contains $M$ codewords. By Theorem 1.11.2, the spheres of radius $t$ about these distinct codewords are disjoint. Define

$$\alpha := \sum_{i=0}^{t} \binom{n}{i} (q-1)^i.$$  

Then, $\alpha$ is the total number of vectors. Then, $M \alpha$ cannot be bigger than the number $q^n$ of vectors in $\mathbb{F}_q^n$. Hence, we must have

$$M \alpha \le q^n$$

or

$$B_q(n, d) \le A_q(n, d) \le \frac{q^n}{\alpha}$$

which is precisely the sphere packing bound. \qed

**Definition 1.11.3.** Let $C$ be a $[n, k, d]_q$ code and $t = \lfloor \frac{d-1}{2} \rfloor$. If the spheres of radius $t$ are pairwise disjoint and their union is the entire space $\mathbb{F}_q^n$, then the code $C$ is said to be perfect.

**Example:** 1.12.2 in the book.

We know that $\mathcal{H}_{q,r}$ over $\mathbb{F}_q$ is an $[n, k, 3]$ code where $n = (q^r-1)/(q-1)$ and $k = n-r$ ($t = 1$). Then,

$$\frac{q^n}{\sum_{i=0}^{t} \binom{n}{i} (q-1)^i} = \frac{q^n}{1 + n(q-1)} = \frac{q^n}{q^r} = q^k$$

Hence, the Hamming codes are perfect.

**Theorem 1.11.4.**  
(i) There exist perfect single error-correcting codes over $\mathbb{F}_q$ which are not linear and all codes have parameters corresponding to Hamming codes.

(ii) The only non-trivial perfect multiple error-correcting codes have the same length, number of codewords, and minimum distance as either the $[23, 12, 7]$ Golay code or the $[11, 6, 5]$ ternary Golay code.
(iii) Any binary possibly nonlinear code with \(2^{12}\) (respectively \(3^6\)) vectors containing the 0 vector with length 23 (resp. 11) and minimum distance 7 (resp. 5) is equivalent to the \([23,12,7]\) binary (resp. \([11,6,5]\) ternary) Golay code.

Definition 1.11.5. The covering radius, \(\rho(C)\) (linear code) is the smallest integer \(s\) so that \(\mathbb{F}_q^n\) is the union of spheres with radius \(s\) centered at codewords. Equivalently,

\[
\rho(C) = \max_{x \in \mathbb{F}_q^n} \min_{c \in C} d(x, c)
\]

Note that \(\rho(C) \geq t\) and \(\rho(C) = t\) if and only if \(C\) is perfect.

Definition 1.11.6. We say that \(C\) is quasi-perfect if \(\rho(C) = t + 1\).

Theorem 1.11.7. Let \(C\) be linear and \(H\) a parity check matrix.

(i) \(\rho(C)\) is the weight of the coset of largest weight.

(ii) \(\rho(C)\) is the smallest number such that every nonzero syndrome is a combination of \(s\) or fewer columns of \(H\), i.e., there exists a syndrome requiring \(s\) columns.

Theorem 1.11.8. Let \(C = [n,k]_q\) code, \(\hat{C}\) the extension of \(C\), and \(C^*\) be the puncturing of \(C\) on any coordinate. Then,

(i) \(C = C \oplus C_2 \iff \rho(C) = \rho(C_1)\rho(C_2)\).

(ii) \(\rho(C^*)\) is either \(\rho(C)\) or \(\rho(C) - 1\).

(iii) \(\rho(C)\) is either \(\rho(C)\) or \(\rho(C) + 1\).

(iv) If \(q = 2\), then \(\rho(C) = \rho(C) + 1\).

(v) Assume \(x\) is a coset leader of \(C\). If \(x' \in \mathbb{F}_q^n\), all of whose nonzero entries agree with \(x\), then \(x'\) is also a coset leader of \(C\). In particular, if there exists a coset with weight \(s\), there exists a coset of any weight less than \(s\).

Proof. Part (iv). Let \(x = (x_1, \ldots, x_n)\) be a coset leader; then define \(x' = (x_1, \ldots, x_n, 1)\). It is enough to show that \(x'\) is a coset leader. Let \(c = (c_1, \ldots, c_n) \in C\) and \(\hat{c}\) be its extension.

If the weight of \(c\) is even, then

\[
\text{wt}(\hat{c} + x') = \text{wt}(c + x) + 1 \geq \text{wt}(x) + 1
\]

where the last inequality is because \(x\) is a coset leader (\(\text{wt}(x) \leq \text{wt}(x + c)\) for all codewords).

If the weight of \(c\) is odd, then

\[
\text{wt}(\hat{c} + x') = \text{wt}(c + x)
\]

By Theorem 1.4.3, we get that the \(\text{wt}(c + x)\) is odd if and only if \(\text{wt}(x)\) is even. In particular, the \(\text{wt}(c + x) \neq \text{wt}(x)\). Therefore,

\[
\text{wt}(c + x) > \text{wt}(x)
\]

and

\[
\text{wt}(\hat{c} + x') = \text{wt}(c + x) \geq \text{wt}(x) + 1
\]

Thus, the

\[
\text{wt}(x') = \text{wt}(x) + 1 \leq \text{wt}(\hat{c} + x')
\]

for all \(\hat{c} \in \hat{C}\). Hence, \(x'\) is a coset leader.
Example 1.12.7: Let $C$ be generated by $G = [1, 1, 2]$. Then,

$$C = \{000, 112, 221\}$$

$d = 3$, $t = 1$.

$$|B_1(c)| = \sum_0^1 \left( \begin{array}{c} 3 \\ i \end{array} \right) 2^i = 1 + 6 = 7$$

However, let $(x_1, x_2, x_3) \in \mathbb{F}_3^3$. Note that each vector is less than two away from an element of $C$, so $\rho(C) = 2$.

Now, let’s consider the extension of $C$, $\hat{C}$. This is generated by $\hat{G} = [1122]$:

$$C = \{0000, 1122, 2211\}$$

Here, $d = 4$ and $t = 1$. We can tell that $\rho(\hat{C}) > 1$ because $\rho(C) = 2$. Suppose that $(x_1, x_2, x_3, x_4)$ is not within 2 of 0000 and 1122. By exhaustion, we can see that this cannot happen.

Wednesday, 6-15-2005:

2. Bounds on the size of Codes

Consider a linear code with $[n, k, d]_q$. We want both $k$ and $d$ large compared to $n$. We want to pay attention to the information rate, $R = \frac{k}{n}$ and the relative minimum distance $\delta = \frac{d}{n}$. Note $0 \leq R \leq 1$ and $0 \leq \delta \leq 1$.

$$\alpha - q(\delta) = \limsup_{n \to \infty} \{R : \exists [n, k, d]_q \text{code with } \frac{k}{n} = R, \frac{d}{n} = \delta\}$$

What is $\alpha_q(\delta)$? No one knows. However, we do know that this is a continuous function of $\delta$, and we have various bounds -

- Plotkin - $\alpha_q(\delta)$ lies below a line from $(1, 0)$ to $(0, 1 - 1/q(\delta) \text{ on } (R, \delta) \text{ axis}$. Discovered early on.
- Gilbert-Varshamov Bound - $\alpha_q(\delta)$ lies above the curve. Constructive bound. Discovered early on.
- People though that Gilbert-Varshamov Bound was actually $\alpha_q(\delta)$ until about 1980, when Goppa introduced algebraic geometry codes.
- In 1982, Tsfasman, Vladut, Zink provide that there exists a sequence of algebraic geometry codes which beats the Gilbet-Varshamov bound.

2.1. $A_q(n, d)$ and $B_q(n, d)$.

Let $C$ be a $(n, M, d)_q$ code. This notation means that $C \subset \mathbb{F}_q^n$, possibly nonlinear, the number of codewords is $M$, and

$$d(x, y) \geq d, \forall x, y \in C, x \neq y$$

where there exists a $x, y \in C$ with $d(x, y) = d$.

If $C$ is an $[n, k, d]_q$ code, then $C$ is an $(n, q^k, d)_q$ code.

Notation:

$$A_q(n, d) = \max \{M : \exists (n, M, \geq d)_q \text{code}\}$$
Proof. \( B_q(n, d) = \max \{ q^k : \exists [n, k, d]_q \text{code} \} \)

Of course, \( B_q(n, d) \leq A_q(n, d) \). We can do some computations easily -

\[ B_q(n, 1) = q^n \]

\[ A_q(n, 1) = q^n \]

**Theorem 2.1.1.** Assume \( d > 1 \). Then,

1. \( A_q(n, d) \leq A_q(n - 1, d - 1) \)
2. If \( q = 2 \) and \( d \) is even, \( A_q(n, d) = A_q(n - 1, d - 1) \)
3. (1), (2) and (4) hold if we replace all \( A \)’s with \( B \)’s.
4. If \( q = 2 \) and \( d \) is even, set \( M = A_{2}(n, d) \). Then there exists \( (n, M, \geq d) \) code such that \( wt(x) \) is even for all \( x \in \mathcal{C} \) and \( \langle x, y \rangle \) is even for all \( x, y \in \mathcal{C} \).

**Proof.**

1. Set \( M = A_q(n, d) \) and let \( \mathcal{C} \) be an \( (n, M, \geq d) \) code. Puncture \( \mathcal{C} \) on any coordinate to get \( \mathcal{C}' \), \( (n - 1, M, \geq d - 1) \) code. So \( A_q(n - 1, d - 1) \geq M = A_q(n, d) \).
2. Set \( M = A_{2}(n - 1, d - 1) \) and assume that \( d \) is even. Let \( \mathcal{C} \) be an \( (n - 1, M, \geq d - 1) \) code. Then \( \mathcal{C} \) is an \( (n, M, \geq d) \) code so \( A_{2}(n, d) \geq M = A_{2}(n - 1, d - 1) \). By part 1, we get the other inequality.
3. Puncturing or extending a linear code yields a linear code.
4. Let \( \mathcal{C} \) be an \( (n, M, \geq d) \) code. Puncture to get \( \mathcal{C}' \), \( (n - 1, M, d - 1) \) code. \( d - 1 \) is odd, so \( \mathcal{C}' \) is an \( (n, M, d) \) code and every vector in \( \mathcal{C}' \) has even weight.

\[ \square \]

**Theorem 2.1.2.** \( A_{2}(n, 2) = B_{2}(n, 2) = 2^{n-1} \).

**Proof.** \( A_{2}(n, 2) = A_{2}(n - 1, 1) = 2^{n-1} \).

Let \( \mathcal{C} \) be the binary code of length \( n \) consisting of all even-weight vectors. This is a linear code with minimum distance \( d_{\min}(\mathcal{C}) = 2 \) with dimension \( n - 1 \) (the first \( n - 1 \) coordinates determine the last one). So, \( \mathcal{C} \) is an \( [n, n - 1, 2] \) code and \( B_{2}(n, 2) \geq 2^{n-1} \). But, \( B_{2}(n, 2) \leq A_{2}(n - 1, 1) \), so \( A_{2}(n, 2) = B_{2}(n, 2) = 2^{n-1} \).

**Remark:** For arbitrary \( q \), we have

\[ q^{n-1} \leq B_q(n, 2) \]

by taking the full even-like subcode of \( \mathbb{F}_q^n \).

\[ B_q(n, 2) \leq A_q(n, 2) \leq A_q(n - 1, 1) = q^{n-1} \]

Hence, the above theorem holds for arbitrary \( q \).

**Theorem 2.1.3.** \( A_q(n, n) = B_q(n, n) = q \).

**Proof.** The repetition code is an \( [n, 1, n]_q \) code. So, \( q \leq B_q(n, n) \leq A_q(n, n) \).

Suppose there is a code, \( \mathcal{C} \) (possibly nonlinear which has parameters \( (n, M, n)_q \) with \( M \geq q + 1 \). Then the entry in the first coordinates of each codeword of \( \mathcal{C} \) is chosen from \( \mathbb{F}_q \) (there are \( q \) choices). As there are \( q + 1 \) distinct codewords, there must be two codewords that agree on the first coordinate. But the distance between those two codewords must be strictly less than \( n \), a contradiction. Hence, \( A_q(n, n) = B_q(n, n) = q \).
Theorem 2.1.4.

\[ A_q(n, d) \leq qA_q(n - 1, d) \]

\[ B_q(n, d) \leq qB_q(n - 1, d) \]

The nonlinear case is in the book. For the linear case, let \( q^k = B_q(n, d) \), and let \( C \) be an \([n, k, \geq d]_q \) code. So there exists \( i \) such that not every codeword is 0 on the \( i \)th coordinate. So, \( q^{k-1} \) codewords are 0 on this coordinate. So, shortening our code at this coordinate, we get an \([n - 1, k - 1, \geq d]_q \) code. This says that

\[ B_q(n - 1, d) \geq q^{k-1} = \frac{B_q(n, d)}{q} \]

Corollary 2.1.5. Singleton Bound:

\[ A_q(n, d) \leq q^{n-d+1} \]

for all \( d \leq n \).

Proof. If \( d = n \), \( A_q(n, n) = q = q^{n-n+1} \). If \( d < n \), then

\[ A_q(n, d) \leq qA_q(n - 1, d) \leq q^2A_q(n - 2, d) \leq \cdots \leq q^{n-d}A_q(d, d) = q^{n-d+1} \]

\[ \square \]

Thursday, 16 June 2005:

2.2. Plotkin Bound.

Theorem 2.2.1. Let \( r = 1 - \frac{1}{q} \). Then,

\[ A_q(n, d) \leq \left\lfloor \frac{d}{d - rn} \right\rfloor \]

if \( n \leq d \leq rn \).

Proof. Let \( M = A_q(n, d) \). Let \( C \) be an \((n, M, \geq d)_q \) code. Let

\[ S = \sum_{x \in C} \sum_{y \in C} d(x, y) \]

We will find an upper bound and a lower bound for \( S \).

Lower Bound:

\[ d(x, y) = 0, \ x = y \]

\[ d(x, y) \geq d, \text{ otherwise} \]

\[ S \geq M(M - 1)d. \]

Upper Bound:

For \( 1 \leq i \leq n \) and \( \alpha \in \mathbb{F}_q \), set

\[ n_{i, \alpha} = \# \{ x \in C : x_i = \alpha \} \in \mathbb{Z} \]

\[ S = \sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} n_{i, \alpha}(M - n_{i, \alpha}) \]
\[
= M \sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha} - \sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha}^2
\]
\[
= nM^2 - \sum_{i=1}^{n} \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha}^2
\]

From Cauchy-Schwarz, with
\[
a = (n_{i,\alpha})_{\alpha \in \mathbb{F}_q}
\]
\[
b = (1, \ldots, 1) \in \mathbb{R}^q
\]
\[
\sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha} \leq \left( \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha}^2 \right)^{1/2} \sqrt{q}
\]

So,
\[
\frac{1}{q} \left( \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha} \right)^2 \leq \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha}^2
\]

Returning to our original equality,
\[
S \leq nM^2 - \frac{n}{q} \left( \sum_{\alpha \in \mathbb{F}_q} n_{i,\alpha} \right)^2
\]

Putting everything together,
\[
dM(M-1) \leq S \leq nM^2 - \frac{n}{q} M^2
\]
\[
dM - d \leq nM - \frac{n}{q} M = nM(1 - \frac{1}{q}) = nrM
\]
\[
M \leq \left\lfloor \frac{d}{d-rn} \right\rfloor
\]

**Remarks:**

1. If \( q = 2 \), the Plotkin bound can be improved (p. 59). If \( n < 2d \), then

\[
A_2(n, d) \leq 2 \left\lfloor \frac{d}{2d-n} \right\rfloor
\]

2. Asymptotic Plotkin Bound says: Let \( r = 1 - \frac{1}{q} \). Then,

\[
\alpha_q(\delta) \leq 1 - \frac{\delta}{q}, \text{ if } 0 \leq \delta \leq r
\]

\[
\alpha_q(\delta) = 0, \text{ if } r < \delta \leq 1
\]

The proof, for \( r < \delta \leq 1 \) is to apply the usual Plotkin Bound. For \( 0 \leq \delta \leq r \), shorten enough so that usual Plotkin applies.

3. \( A_2(2d, d) \leq 4d \) if \( d \) is even. \( A_2(2d, d) \leq 2d + 2 \) if \( d \) is odd. \( A_2(2d+1, d) \leq 4d + 4 \) if \( d \) is odd. These can be shown by stretching the Plotkin Bound with the simple facts learned in the previous section.
2.3. The Nordstrom Robinson Code. Demonstrably better than binary linear codes. Is a \((16, 256, 6)\) nonlinear code. The construction uses the extended Golay code, \(G_{24}\).

Let \(C\) be a code equivalent to the Golay code \(G_{24}\), with \(c = [1^8|0^1|6] \in C\). \(C\) is a \([24,12,8]\) self-dual code.

Let \(T = \{1, \ldots, 8\}\), \(T^1 = \{9, \ldots, 24\}\),

\[C(T) = \{x \in C : x_i = 0 \forall i \in T\},\]

\(C_T = C\) shortened on \(T = C(T)\) punctured on \(T\)

\[C_T' = C\text{ punctured on } T'\]

Claim: \(C_T'\) is the \([8, 7, 2]\) code consisting of all vectors of length 8 having even weight.

Let \(G\) be a generator matrix for \(C\). \(C = C^\perp\), so \(G\) is also a parity check matrix. Since \(d_{\min}(C) = 8\), every \(d - 1 = 7\) columns of \(G\) are linearly independent. So, \(\dim C_T' \geq 7\). Since \(c \in C\), \(\text{wt}(x)\) is even for all \(x \in C_T'\). So \(C_T'\) is as desired.

Claim: \(C_T\) is \([17,5,\geq 8]\) code.

Proof: \(C = C^\perp\), \(C_T = (C^\perp)_T\). Therefore, from a previous theorem, \(\dim C_T = n - d - k + 1 = 5\). Then,

\[d_{\min}(C_T) = d_{\min}(C(T)) \geq d_{\min}(C) = 8\]

Since every even weight vector of length 8 is in \(C_T'\), we can find \(c^1, \ldots, c^7 \in C\) such that \(c^i\) has 1 in the \(i\)th coordinate, 1 in the 8th coordinate, and 0’s elsewhere for coordinates less than 8. Set \(c^0 = 0 \in \mathcal{C}\). For \(0 \leq i \leq 7\), set

\[\mathcal{C}_i = c^i + \mathcal{C}(T) \subset \mathcal{C}\]

Each of the \(\mathcal{C}_i\) is disjoint, and each \(\mathcal{C}_i\) has size \(|\mathcal{C}(T)| = 32\).

Set

\[\mathcal{N} = \mathcal{C}_0 \cup \cdots \cup \mathcal{C}_7 \subset \mathcal{C}\]

\[\mathcal{N}_{16} = \mathcal{N}^T \subset C^T\]

So,

\[\mathcal{N}_{16} = \mathcal{C}_0^T \cup \cdots \mathcal{C}_7^T\]

is the Nordstrom-Robinson code.

\[\#\mathcal{N}_{16} = \mathcal{N} = 8 \cdot 32 = 256\]

Suppose \(a, b \in \mathcal{N} \subset \mathcal{C}\). Then,

\[d(a, b) \geq 8\]

and at most 2 of the coordinates on which they differ are in \(T\). So, when we puncture they still differ on \(\geq 6\) coordinates. So, \(d_{\min}(\mathcal{N}_{16}) \geq 6\).
2.4. The Griesmer Upper Bound. Serves as a generalization of the Singleton Bound. Applies only to linear codes.

We start out with the idea of a residual code.

**Definition 2.4.1.** Let $C$ be an $[n, k]$ code and let $c$ be a codeword of weight $w$. Denote the set of nonzero coordinates of $c$ as $I$. The **residual code of $C$ with respect to $c$** is the code of length $n - w$ punctured on $I$.

**Notation:** This is denoted as $\text{Res}(C, c)$.

Question of the moment: How does the original code relate to the residual one?

**Theorem 2.4.2.** Let $C$ be an $[n, k, d]$ code over $\mathbb{F}_q$ and let $c$ be a code of weight $w < qd - (q - 1)$. Then $\text{Res}(C, c)$ is an $[n - w, k - 1, d']$ code, where $d' \geq d - w + \lceil w/q \rceil$.

**Proof.** By replacing $C$ with a monomially equiv. code, we can assume WLOG that $c = (1, \ldots, 1, 0, \ldots, 0)$ (with $w$ 1’s). Since puncturing $c$ on its nonzero coordinates gives the zero vector, $\text{Res}(C, c)$ has at most dimension $k - 1$. Assume the dimension is strictly less than $k - 1$. Then, there exists a codeword $x = (x_1, \ldots, x_n) \in C$ which is not a multiple of $c$ with $x_{w+1}, \ldots, x_n = 0$.

By a counting argument, there exists $\alpha \in \mathbb{F}_q$ such that at least $w/q$ coordinates of $y_1, \ldots, y_w$ equal $\alpha$. Therefore,

$$d \leq \text{wt}(x - \alpha c) \leq w - \frac{w}{q} = \frac{w(q - 1)}{q}.$$

This contradicts the hypothesis on $w$; $\text{Res}(C, c)$ has dimension $k - 1$.

Let $(y_{w+1}, \ldots, y_n)$ be any nonzero codeword in $\text{Res}(C, c)$, and let $y = (y_1, \ldots, y_w, y_{w+1}, \ldots, y_n)$ be the corresponding codeword in $C$. Similarly to before, there exists a $\alpha \in \mathbb{F}_q$ such that at least $w/q$ coordinates of $y_1, \ldots, y_w$ equal $\alpha$. So,

$$d \leq \text{wt}(y - \alpha c) \leq w - \frac{w}{q} + \text{wt}(y_{w+1}, \ldots, y_n).$$

Rearranging,

$$\text{wt}(y_{w+1}, \ldots, y_n) \geq d - w - \lceil w/q \rceil$$

Because $(y_{w+1}, \ldots, y_n)$ was an arbitrary codeword in $\text{Res}(C, c)$, we get that

$$d' \geq d - w - \lceil w/q \rceil$$

□

If $c$ is a codeword of weight $d$, then we get the following corollary:

**Corollary 2.4.3.** If $C$ is an $[n, k, d]$ code over $\mathbb{F}_q$ and $c \in C$ has weight $d$, then $\text{Res}(C, c)$ is an $[n - d, k - 1, d']$ code, where $d' \geq \lfloor d/q \rfloor$. 
Example: 2.7.3 from the book. Where is this useful? Showing that a code doesn’t exist.
Suppose we have a $[16, 8, 6]$ binary linear code. Pick a minimum weight vector $c$, and consider $C' = \text{Res}(C, c)$. Then, by the previous Corollary, $C'$ is a $[10, 7, d']$ code with $d' \geq \lceil 6/2 \rceil = 3$. By the Singleton Bound, $d' \leq 10 - 7 + 1 = 4$. By Theorem 2.4.4, $d' = 4$ is not possible (I don’t claim any knowledge to this theorem). By the Sphere Packing Bound, if $d' = 3$, then,

$$2^7 < \frac{2^{10}}{\binom{10}{0} + \binom{10}{1}}$$

$$128 < 93,$$
a contradiction. Hence, no $[16, 8, 6]$ binary linear code can exist.

**Theorem 2.4.4. Griesmer Bound**

Let $C$ be an $[n, k, d]$ code over $F_q$ with $k \geq 1$. Then,

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

**Proof.** We would like to show this by induction. If $k = 1$, then the conclusion is trivial (we have an empty sum, so $k = 1$ says $n \geq 0$). Assume for $k > 1$, and assume that $c \in C$ is a codeword of weight $d$. By the previous Corollary, $\text{Res}(C, c)$ is an $[n - d, k - 1, d']$ code, where $d' \geq \lceil d/q \rceil$. Applying the induction hypotheses to $\text{Res}(C, c)$,

$$n - d \geq \sum_{i=0}^{k-2} \left\lceil \frac{d'}{q^i} \right\rceil = \sum_{i=0}^{k-2} \left\lceil \frac{d}{q^i} \right\rceil$$

$$n - d \geq \sum_{i=0}^{k-2} \left\lceil \frac{d}{q^{i+1}} \right\rceil$$

Adding $d$ to both sides and re-indexing the summation,

$$n \geq \sum_{i=0}^{k-1} \left\lceil \frac{d}{q^i} \right\rceil.$$

Because $\lceil d/q^i \rceil \geq 1$ for $i = 1, \ldots, k - 1$ and $\lceil d/q^0 \rceil = d$, the Griesmer Bound implies the linear case of the Singleton Bound.

The Griesmer bound is not always achievable; for example, a binary code of dimension 12 and minimum distance 7 has length $n \geq 22$. However, by another bound in the book (Johnson Bound), implies that $n \geq 23$.

So in what cases do we meet the Griesmer Bound? The most popular ones are the simplex codes.

**Theorem 2.4.5.** Every nonzero codeword of the $r$-dimensional simplex code over $F_q$ has weight $q^{r-1}$. The simplex codes meet the Griesmer Bound.
Proof. Let $G$ be a generator matrix for the $r$-dimensional simplex code $C$ over $\mathbb{F}_q$. The matrix $G$ is formed by choosing for its columns a nonzero vector from each 1-dimensional subspace of $\mathbb{F}_q^r$. Because $C = \{xG : x \in \mathbb{F}_q^r\}$, if $x \neq 0$, then $\text{wt}(G) = n - s$, where $s$ is the number of columns $y$ of $G$ such that $x \cdot y^T = 0$. The set of vectors of $\mathbb{F}_q^r$ orthogonal to $x$ is an $(r - 1)$ dimensional subspace of $\mathbb{F}_q^r$ and thus exactly $(q^{r-1} - 1)/(q - 1)$ columns $y$ of $G$ satisfy $x \cdot y^T = 0$. Thus, $\text{wt}(xG) = (q^{r-1} - (q^{r-1} - 1))/(q - 1) = q^{r-1}$. This shows each nonzero codeword has weight $q^{r-1}$.

Hence, the minimum distance is $q^{r-1}$. Since $r - 1 \sum_{i=0}^{r-1} \left\lceil \frac{d}{2^i} \right\rceil \sum_{i=0}^{r-1} q^i = q^{r-1}$ the simplex codes meet the Griesmer Bound. □

Theorem 2.4.6. Let $C$ be an $[n, k, d]_2$ code that meets the Griesmer Bound. Then $C$ has a basis of minimum weight codewords.

Ideal: Induct on $k$.

Proof. Induct on $k$. $k = 1$ is clear. Assume $k > 1$ and $c \in C$ with $\text{wt}(c) = d$. up to permutation, we have a generator matrix

$$G = \begin{bmatrix} 1 & \ldots & 1 & 0 & \ldots & 0 \\ G_0 & G_1 \end{bmatrix}$$

So, we may assume that $G_1$ is a generator matrix $\text{Res}(C, c)$. It’s an $[n - d, k - 1, d']_2$ code with $d' \geq d_1 := \lceil d/2 \rceil$. Since $C$ meets the Griesmer-Bound, we know

$$n - d = \sum_{i=1}^{k-1} \left\lfloor \frac{d}{2^i} \right\rfloor = \sum_{j=0}^{k-2} \left\lfloor \frac{d_1}{2^j} \right\rfloor$$

Suppose that $d' > d_1$. Then,

$$n - d - \sum_{i=0}^{k-2} \left\lfloor \frac{d'}{2^j} \right\rfloor$$

However, this means that $\text{Res}(C, c)$ violates the Griesmer Bound. So, $d' = d_1$. By induction, we may assume that rows of $G_1$ have weight $d_1$. For $i \geq 2$, define $r_i = (s_{i-1}, t_{i-1})$, where $s_{i-1}$ is the $(i - 1)^{th}$ row of $G_0$ and $t_{i-1}$ is the $(i - 1)^{th}$ row of $G_1$. We claim that either $r_i$ or $c + r_i$ has weight $d$.

□

Notation: $g(k, d) = \sum_{i=0}^{k-1} \left\lfloor \frac{d}{2^i} \right\rfloor$.

If a code meets the Griesmer bound, notice that $n - g(k, d) = 0$ and that $d = d + 0 = d + n - g(k, d)$.

Theorem 2.4.7. Let $C$ be an $[n, k, d]_2$ code and set $h = n - g(k, d)$. Then $C$ has a basis of codewords of weight at most $d + h$. 
Theorem 2.4.8. Suppose that there exists an \([n, k, d]_q\) code \(C\). Then there exists an \([n, k, d]_q\) code \(C’\) such that \(C’\) has a basis of words of weight \(d\).

Proof. Let \(s\) be the maximum number of linearly independent codewords of weight \(d\) in our code \(C\). Clearly, \(s \geq 1\). If \(s = k\), we are done. Suppose \(s < k\), with \(\{e_1, \ldots, e_s\}\) linearly independent codewords of weight \(d\). Then set \(S = \text{span}\{e_1, \ldots, e_s\}\). Then \(C = C \setminus S \cup S\). Note that \(d_{\text{min}}(C \setminus S) > d\). Let \(d_1 = \min\{wt(e) : e \in C \setminus S\}\). So, there exists a vector \(e_1\) with \(wt(e_1) = d_1 > d\). Extend to a basis \(\{e_1, e_2, \ldots, e_{d-1}\}\). Choose \(d_1 - d\) nonzero coordinates of \(e_1\) and form \(e_1'\) which is merely \(e_1\) on the not \(d_1 - d\) chosen coordinates and is 0 on the \(d_1 - d\) coordinates. Then \(wt(e_1') = d\). Let \(C_1 = \text{span}\{e_1, \ldots, e_s, e_1', e_2, \ldots, e_{k-s}\}\). We claim that \(C_1\) is an \([n, k, d]_2\) code. If \(x \in C_1\), then either \(x \in S\) or \(x \in C \setminus S\). If \(x \in S\), then \(wt(x) = d\). If \(x \in C \setminus S\), then \(x\) “came from” some \(y \in C\) so that \(wt(y) \geq d_1\), which implies that \(wt(x) \geq d_1 - (d_1 - d) = d\). Hence \(d_{\text{min}}(C_1) = d\). Suppose \(\dim(C_1) < k\). Then, as \(e_1' \in \text{span}\{e_1, \ldots, e_s, e_2, \ldots, e_{k-s}\}\) \(\subseteq C\), by the maximality of \(s\), we know that \(e_1' \in S\). Then \(e_1 - e_1' \in C \setminus S\), since \(e_1 \notin S\). By construction, \(wt(e_1 - e_1') = d_1 - d\). As \(e_1 - e_1' \in C \setminus S\), we know \(wt(e_1 - e_1') \geq d_1\), which is a contradiction. So \(\dim(C_1) = k\).

\(\square\)

Theorem 2.4.9. Let \(C\) be a linear code over \(\mathbb{F}_p\), where \(p\) is prime, which meets the Griesmer bound. Assume that \(p^n|d\). If \(x \in C\), then \(p^n|\text{wt}(x)\).

6/20/05

2.5. The Gilbert Lower Bound.

Theorem 2.5.1 (Gilbert Bound, ’52).

\[ B_q(n, d) \geq \frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i} = q^{n-d} = q^{n-\log_q \sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i}. \]

Proof. Let \(C\) be a linear code over \(\mathbb{F}_q\) with \(B_q(n, d)\) codewords. By Theorem 2.1.7 the covering radius of \(C\) is at most \(d - 1\). Hence the spheres of radius \(d - 1\) about the codewords cover \(\mathbb{F}_q^n\). By \(V_p(n, a) = \sum_{i=0}^{a} \binom{n}{i}(q-1)^i\), a sphere of radius \(d - 1\) centered at a codeword contains \(\alpha = \sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i\) vectors. As the \(B_q(n, d)\) spheres centered at codewords must fill the space \(\mathbb{F}_q^n\), \(B_q(n, d)\alpha \geq q^n\), giving the bound. \(\square\)

Let \(C\) be a (possibly) non-linear code of dimension \(R\) and minimum distance \(d\) that meets the Gilbert Bound.

1. Start with a vector \(v \in \mathbb{F}_q^n\).
2. Continue to choose vectors in \(\mathbb{F}_q^n\) at least a distance \(d\) from all previous vectors (as long as you can choose them).

The resulting code has minimum distance at least \(d\) (and covering radius at most \(d - 1\)) which meets the Gilbert Bound.

2.6. The Varshamov Lower Bound.

Lemma 2.6.1. Let \(n, k, d \in \mathbb{Z}\) with \(2 \leq d \leq n\) and \(1 \leq k \leq n\) and let \(q\) be a prime power. If

\[ \sum_{i=0}^{d-2} \binom{n-1}{i}(q-1)^i < q^{n-k}, \]
then there exists an \((n - k) \times n\) matrix \(H\) over \(F_q\) so that every set of \(d - 1\) columns of \(H\) is linearly independent.

Proof. From \(F_q^{n-k}\), choose \(h_1 \neq 0\) and \(h_2 \notin \text{span}\{h_1\}\) and in general, \(h_j \notin \text{span}(S)\), where \(S\) is any subset of \(d - 2\) or fewer vectors from \(\{h_1, \ldots, h_j\}\). Eventually, choose \(h_n \notin \text{span}(T)\), where \(T\) is any subset of \(d - 2\) or fewer vectors from \(\{h_1, \ldots, h_n\}\). So every subset of \(\{h_1, \ldots, h_n\}\) of size \(d - 1\) would be linearly independent by our choice of the \(h_i\)'s.

So why can we complete all of the aforementioned steps? We’ll proceed by induction. Let \(j \in \mathbb{Z}\) with \(1 \leq j \leq n - 1\) and assume that \(h_1, \ldots, h_j\) have already been found. Since \(j \leq n - 1\), the number of different linear combinations of \(d - 2\) or fewer of the \(\{h_1, \ldots, h_j\}\) is

\[
\sum_{i=0}^{d-2} \binom{n}{i} (q - 1)^i \leq \sum_{i=0}^{d-2} \binom{n - 1}{i} (q - 1)^i < q^{n-k}.
\]

So by equation 1, we know that there exists a vector \(h_{j+1}\) that we can choose that satisfies everything we wanted it to satisfy.

\[\square\]

Corollary 2.6.2. Let \(n, d, k \in \mathbb{Z}\) with \(2 \leq d \leq n\) and \(1 \leq k \leq n\). Then there exists an \([n, k]\) linear code over \(F_q\) with minimum distance at least \(d\) provided

\[
1 + \sum_{i=0}^{d-2} \binom{n - 1}{i} (q - 1)^i \leq q^{n-k}.
\]

Theorem 2.6.3 (Varshamov Bound, ’57).

\[B_q(n,d) \geq q^n - \left\lfloor \log_q \left(1 + \sum_{i=0}^{d-2} \binom{n - 1}{i} (q - 1)^i\right) \right\rfloor.\]

Proof. Let \(L = \star\) from equation 2. By Corollary 2.6.2, there exists an \([n, k]\) code over \(F_q\) with minimum weight at least \(d\) provided

\[\log_q(L) \leq n - k \iff k \leq n - \log_q(L),\]

and the largest \(k\) is when \(k = n - \left\lfloor \log_q(L) \right\rfloor\). Thus, \(B_q(n,d) \geq q^n - \left\lfloor \log_q(L) \right\rfloor\).

\[\square\]

2.7. Asymptotic Bound. These are the bounds that we get using our previous bounds and then letting the length go to \(\infty\).

Definition 2.7.1. For a possibly non-linear code over \(F_q\) with \(M\) codewords, the information rate is given by

\[n^{-1} \log_q(M)\]

Consider \([n, k, d]\) code with \(M = q^k\). Then the information rate is \(n^{-1} \log_q(q^k) = \frac{k}{n}\), just as before.

In the linear case, the rate is a measure of the number of information coordinates relative to the total number of coordinates.

Definition 2.7.2. If \(C\) is an \([n, k, d]\) code, then \(\frac{d}{n}\) is called the relative distance.

The relative distance is a measure of the error correcting capability of the code relative to its length.
In general, asymptotic bounds will be lower/upper bounds on the largest possible rate for a family of codes over $\mathbb{F}_q$ as lengths go to $\infty$ with relative distance $\delta$. We define the function

$$\alpha_q(d) = \limsup_{n \to \infty} n^{-1} \log_q A_q(n, \delta n).$$

**Theorem 2.7.3** (Asymptotic Gilbert-Varshamov Bound, ’52). If $0 < \delta < 1 - q^{-1}$ with $q \geq 2$, then we find that

$$\alpha_q(\delta) \geq 1 - H_q(\delta),$$

where

$$H_q(\delta) = \begin{cases} 0, & \text{if } x = 0 \\ x \log_q (q - 1) - x \log_q x - (1 - x) \log_q (1 - x), & \text{if } 0 < x \leq r. \end{cases}$$

**Proof.** Using the Gilbert Bound, we know that

$$A_q(n, d) = A_q(n, \lfloor \delta n \rfloor) \geq q^n \frac{V_q(n, \lfloor \delta n \rfloor)}{V_q(n, \lfloor \delta n \rfloor - 1)},$$

where $V_q(n, d)$ is the number of vectors in a sphere of radius $d - 1$ in $\mathbb{F}_q^n$. But, $\lfloor \delta n \rfloor - 1 \leq \lfloor \delta n \rfloor$, and so

$$A_q(n, \delta n) \geq q^n \frac{V_q(n, \lfloor \delta n \rfloor)}{V_q(n, \lfloor \delta n \rfloor)}.$$

So we have

$$\alpha_q(d) = \limsup_{n \to \infty} n^{-1} \log_q A_q(n, \delta n)$$

$$\geq \limsup_{n \to \infty} n^{-1} \log_q \left( q^n \frac{V_q(n, \lfloor \delta n \rfloor)}{V_q(n, \lfloor \delta n \rfloor)} \right)$$

$$= 1 - H_q(\delta).$$

□

### 3. The Theory of Finite Fields

#### 3.1. Finite Fields.
Here are some basic facts.

1. There exists a field with $q$ elements if and only if $q = p^m$ for $p$ a prime and $m \geq 1$.
2. If $q = p^m$, then there is a unique field with $q$ elements (up to isomorphism). We denote this field by $\mathbb{F}_q$.
3. If $q = p^m$, then $\mathbb{F}_q$ has characteristic $p$; i.e., $pa = 0$ for every $a \in \mathbb{F}_q$ and for every $a \in \mathbb{Z}$ with $1 \leq a \leq p - 1$, there exists $\alpha \in \mathbb{F}_q$ with $a\alpha \neq 0$.
4. For all $\alpha, \beta \in \mathbb{F}_q$ (with $q = p^m$) and all $r \geq 1$, then $(\alpha + \beta)^p^r = \alpha^{p^r} + \beta^{p^r}$.

**Proof.** If $r = 1$, then $(\alpha + \beta)^p = \sum_{i=0}^{p} \binom{p}{i} \alpha^{p-i} \beta^i$. As $\binom{p}{i} = \frac{p^i}{i!(p-i)!}$, and since $p$ is prime and $i < p$ and $p - i < p$, then if $1 \leq i \leq p - 1$, then $gcd(p, i!(p-i)!) = 1$. So $p\binom{p}{i}$ for $1 \leq i \leq p-1$. So $(\alpha + \beta)^p = \binom{p}{0} \alpha + \binom{p}{p} \beta^p = \alpha^p + \beta^p$. 

6/21/05
Now assume that \( r > 1 \) and the statement holds for \( r - 1 \). Then

\[
(\alpha + \beta)^{pr} = \left( (\alpha + \beta)^{p^{r-1}} \right)^p \\
= (\alpha^{p^{r-1}} + \beta^{p^{r-1}})^p \\
= (\alpha^{p^{r-1}})^p + (\beta^{p^{r-1}})^p \\
= \alpha^{p^r} + \beta^{p^r}.
\]

\( \square \)

Note that in \( F_2 \), we have

\[
(\alpha + \beta)^6 = ((\alpha + \beta)^2)^3 \\
= \alpha^6 + 3\alpha^4\beta^2 + 3\alpha^2\beta^4 + \beta^6 \\
= \alpha^6 + \alpha^4\beta^2 + \alpha^2\beta^4 + \beta^6 \\
\neq \alpha^6 + \beta^6.
\]

So this means that Freshmen’s dream only holds for powers of a prime, not merely a product of the prime.

(5) \( F_q[x] \) is a Euclidean Domain (hence a PID and UFD).
(6) \( (x - \alpha)|f(x) \iff f(\alpha) = 0. \)
(7) \( \deg(f(x), g(x)) = \deg(f(x)) + \deg(g(x)). \)
(8) \( f(x) \in F_q[x] \) has at most \( \deg(f(x)) \) roots in any field containing \( F_q \).
(9) For every \( m \geq 1 \), there is at least one monic irreducible polynomial of degree \( m \) in \( F_p[t] \) for \( p \) prime. If \( f(t) \in F_p[t] \) is irreducible and monic of degree \( m \), then \( F_p[t]/f(t) \cong F_{p^m} \).
(10) \( F_q = \{ x \text{ in any field containing } F_q : x^q - x = 0 \} \).
(11) \( F_{p^r} \subseteq F_{p^s} \iff r|s. \) (For example, \( F_4 \subseteq F_{16} \) but \( F_4 \not\subseteq F_8 \) and \( F_8 \not\subseteq F_{16} \).)
(12) \( F_q^* \) is a cyclic group of order \( q - 1 \). Any generator is called a \textit{primitive element} of \( F_q \). The number of primitive elements if \( \phi(q - 1) = \#\{ a : 1 \leq a \leq q - 1, \gcd(a, q - 1) = 1 \} \). If \( \gamma \) is primitive, then \( \gamma^a \) is primitive if and only if \( \gcd(a, q - 1) = 1 \). In general, the order of \( \gamma^a \) is \( \frac{q - 1}{\gcd(a, q - 1)} \).
(13) If \( \sigma \) is an automorphism of \( F_q \) with \( q = p^m \), then \( \sigma(a) = a \) for every \( a \in F_p \).

\textit{Proof.} \( \sigma(1) = 1 \), and so \( \sigma\left(\underbrace{1 + \cdots + 1}_{a}\right) = \underbrace{1 + \cdots + 1}_{a} = a. \)

\( \square \)

(14) The map \( \sigma_p : F_q \to F_q \) given by \( \sigma_p(\alpha) = \alpha^p \) is an automorphism of \( F_q \). This is called the Frobenius automorphism.
(15) \( \{ \sigma : F_q \to F_q : \sigma \text{ is an automorphism} \} \) is a group under composition. This is the Galois group of \( F_q \) over \( F_p \).
(16) \( \text{Gal}(F_q/F_p) \) is cyclic of order \( m \), where \( q = p^m \) and is generated by \( \sigma_p \), the Frobenius automorphism.

\textbf{Example 3.1.1.} Consider the field \( F_{16} \). To construct \( F_{16} \) over \( F_2 \), we need an irreducible polynomial in \( F_2[t] \). Let \( f(t) = t^4 + t + 1 \). As there are no roots, we see that if \( f(t) \) is not irreducible, then \( f(t) = (g(t))^2 \), where \( g(t) = t^2 + t + 1 \), the unique irreducible polynomial of degree 2. As \( (g(t))^2 = (t^2 + t + 1)^2 = t^4 + t^2 + 1 \neq t^4 + t + 1 \).

\( \text{Hence, } f(t) \text{ is irreducible. So} \)

\( F_2[t]/(t^4 + t + 1) \)
is a field with 16 elements. Let \( \alpha \) be the image of \( t \) in the quotient. Then

\[
F_{16} = \{ a + b \alpha + c \alpha^2 + d \alpha^3 : a, b, c, d \in F_2 \}.
\]

Addition is modulo two, treating \( \alpha \) as a formal symbol. Multiplication is modulo two where we replace \( \alpha^4 \) with \( \alpha + 1 \). Multiplicatively, we have

\[
\begin{align*}
0, 1, \alpha, \alpha^2, \alpha^3, \alpha^4 &= \alpha + 1, \\
\alpha^5 &= \alpha^2 + \alpha, \\
\alpha^6 &= \alpha^3 + \alpha^2, \\
\alpha^7 &= \alpha^3 + \alpha + 1, \\
\alpha^8 &= \alpha^2 + 1, \\
\alpha^9 &= \alpha^3 + \alpha,
\end{align*}
\]

\[
\alpha^{10} = \alpha^2 + \alpha + 1, \\
\alpha^{11} = \alpha^3 + \alpha^2 + \alpha, \\
\alpha^{12} = \alpha^3 + \alpha^2 + \alpha + 1, \\
\alpha^{13} = \alpha^3 + \alpha^2 + 1, \\
\alpha^{14} = \alpha^3 + 1.
\]

Note that the primitive elements are \( \{ \alpha, \alpha^2, \alpha^4, \alpha^7, \alpha^8, \alpha^{11}, \alpha^{13}, \alpha^{14} \} \).

\[
6/22/05
\]

3.2. Cyclotomic Cosets and Minimal Polynomials. Let \( E = F_{q^r} \) be an extension of \( F_q \). Let \( \alpha \in E \) and suppose that \( \alpha^q - \alpha = 0 \).

**Definition 3.2.1.** The monic polynomial \( M_\alpha(x) \) in \( F_q[x] \) of smallest degree which has \( \alpha \) as a root is called the minimal polynomial of \( \alpha \) over \( F_q \).

**Theorem 3.2.2.**
1. \( M_\alpha(x) \) is irreducible over \( F_q \).
2. If \( g(x) \in F_q[x] \) with \( g(\alpha) = 0 \), then \( M_\alpha(x) | g(x) \).
3. \( M_\alpha(x) \) is unique.

**Theorem 3.2.3.** Let \( f(x) \) be a monic irreducible polynomial in \( F_q[x] \) of degree \( r \). Then

1. All the roots of \( f(x) \) are in \( F_{q^r} \) and in any field containing \( F_q \) along with one root of \( f(x) \).
2. \( f(x) = \prod_{i=1}^r (x - \alpha_i) \) where the \( \alpha_i \in F_{q^r} \) are distinct for \( 1 \leq i \leq r \).
3. \( f(x) \in F_q[x] \) is irreducible over \( F_q \).

**Proof.**
1. Let \( \alpha \) be a root of \( f(x) \) and make the smallest field extension \( E_\alpha \) containing \( \alpha \) and \( F_q \). If \( \beta \) is another root of \( f(x) \), then form the field extension \( E_\beta \). Finally, form \( E_{\alpha, \beta} \) the smallest field extension of \( E_\alpha \) and \( E_\beta \). Then note that \( |E_\alpha| = |E_\beta| = q^r \). A theorem tells us that any two subfields of a common field with the same cardinality are equal. Hence, \( E_{\alpha, \beta} = E_\alpha \).

**Theorem 3.2.4.** Let \( F_{q^r} \supseteq F_q \) for \( \alpha \in F_{q^r} \). Then

1. \( M_\alpha(x) | x^{q^r} - x \).
2. \( M_\alpha(x) \) has distinct roots in \( F_{q^r} \).
3. \( \deg(M_\alpha(x)) | t \).
4. \( x^{q^r} - x = \prod_{\alpha} M_\alpha(x) \); the \( M_\alpha(x) \)'s run over distinct minimal polynomials.
5. \( x^{q^r} - x = \prod_f f(x) \), where \( f \) runs over distinct monic irreducible polynomials over \( F_q \) whose degree divides \( t \).

**Theorem 3.2.5.** Let \( f(x) \in F_q[x] \) and \( \alpha \) a root of \( f(x) \). Then

1. \( f(x^q) = \sum_{i=0}^n f_i x^{q^i} = \sum_{i=0}^n (f_i x^q)^i = (\sum_{i=0}^n f_i x^i)^q = (f(x))^q \).
2. \( f(\sigma_p^m(\alpha)) = f(\alpha^q) = (f(\alpha))^q = 0 \); i.e., \( \sigma_p^m \) permutes the roots of \( f(x) \).

In other words, we have \( \langle (\sigma_p^m) \rangle = \text{Gal}(F_{q^r} : F_q) \). If we take \( \alpha \in F_{q^r} \), then

\[
\{ \sigma_p^m(\alpha), (\sigma_p^m)^2(\alpha), \ldots \}
\]

is the set of all roots of \( f(x) \). So we know that \( \prod(x - \alpha^q) | M_\alpha(x) \).

**Theorem 3.2.6.** \( \prod(x - \alpha^{q^r}) = M_\alpha(x) \).
Let $\gamma \in F_{q^r}$ be a primitive element. Then $\alpha = \gamma^s$. Then we need to compute \( \{\gamma^s, \gamma^{sq}, \gamma^{sq^2}, \ldots, \gamma^{sq^{r-1}}\} \), where \( r \) is the first positive integer so that \( sq^r \equiv s \pmod{q^r-1} \). We can produce this list of exponents without knowing about \( \gamma \). We can form the set\[ C_s = \{s, sq, sq^2, \ldots, sq^{r-1}\}, \]
which we call the \( q \)-cyclotomic coset of \( s \) modulo \( q^r-1 \).

**Exercise 3.2.7.** Compute the 2-cyclotomic cosets modulo 
\[
\begin{align*}
(a) & \quad C_0 = \{0\}, & C_1 = \{1, 1 \cdot 2, 2 \cdot 2\}, & C_2 = \{3, 6, 5\}.
\end{align*}
\]

**Proof.** (a) \( C_0 = \{0\}, \) then \( M_{\gamma^3}(x) = (x - \gamma^3)(x - \gamma^6)(x - \gamma^5) \).

\[ \square \]

4. Cyclic Codes

4.1. Factoring \( x^n - 1 \).

**Definition 4.1.1.** A code \( C \) is cyclic if \( C \) is linear and if \( c_0c_1 \ldots c_{n-1} \in C \), then \( c_{n-1}c_0 \ldots c_{n-2} \in C \).

Consider \( c_0c_1 \ldots c_{n-1} \in C \) as \( c_0 + c_1x + \cdots + c_{n-1}x^{n-1} \in F_q[x] \). Then we want \( c_0x + c_1x^2 + \cdots + c_{n-1}x^n \) to correspond to \( c_{n-1}c_0 \ldots c_{n-2} \), and so we consider these polynomials in \( R_n := F_q[x]/x^n - 1 \). If \( I \) is an ideal of \( R_n \), then \( I \) corresponds to \( J \subseteq F_q[x] \) such that \( x^n - 1 \in J \). Assume that \( gcd(n, p) = 1 \). Then \( x^n - 1 \) has no repeated roots. We want to find a \( t \) so that \( x^n - 1 \) factors into linear terms in \( F_q[x] \). If \( \alpha, \beta \) are roots, then \( (\alpha\beta)^n - 1 = \alpha^n\beta^n - 1 = 1 \cdot 1 - 1 = 0 \). We need \( \alpha, \beta \) to be roots at \( \{\alpha^s, \alpha^{sq}, \ldots, \alpha^{sq^{r-1}}\} \), where \( r \) is the smallest integer so that \( sq^r \equiv s \pmod{n} \). In fact \( M_{\alpha^s}(x) = \prod_{i \in C_s} (x - \alpha^i) \), where \( C_s = \{s, sq, \ldots, sq^{r-1}\} \).

Then \( x^n - 1 = \sum_{s} M_{\alpha^s}(x) \), where \( s \) ranges over representatives for the \( q \)-cyclotomic cosets.

**Theorem 4.1.2.** \( |C_s| \) divides \( ord_n(q) \), and \( |C_1| = ord_n(q) \).

4.2. Basic Theory of Cyclic Codes. Cyclic codes over \( F_q \) correspond precisely with the ideals of \( R_n = F_q[x]/x^n - 1 \), which also corresponds precisely to principal ideals of \( R_n \).

Let \( c(x) \in F_q[x] \) and \( c(x) \in R_n \) and assume that \( p \nmid n \).

**Theorem 4.2.1.** Let \( C \) be a nonzero cyclic code that is a subset of \( R_n \). Then there exists \( g(x) \sum_{i=0}^{n-k} g_i x^i \) with \( K := n - deg(g(x)) \) so that
\[
\begin{align*}
(1) & \quad g(x) \text{ is the unique monic polynomial of minimal degree in } C \\
(2) & \quad C = \{g(x)\} \\
(3) & \quad g(x)|(x^n - 1) \\
(4) & \quad \dim C = K \text{ and } \{g(x), xg(x), \ldots, x^{k-1}g(x)\} \text{ is a basis for } C
\end{align*}
\]

6/23/05
(5) If \( c(x) \in C \), then \( c(x) \) is uniquely expressible as \( g(x)f(x) \) so that \( f(x) = 0 \) or \( \deg(f(x)) < K \).

(6) \[
G = \begin{bmatrix}
g_0 & g_1 & \ldots & g_{n-k} & 0 & \ldots & 0 \\
0 & g_0 & g_1 & \ldots & g_{n-k} & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & g_0 & g_1 & \ldots & g_{n-k} \\
\end{bmatrix}
= \begin{bmatrix}
g(x) \\
xg(x) \\
\vdots \\
x^{k-1}g(x) \\
\end{bmatrix}
\]
generates \( C \).

(7) If \( \alpha \) is a primitive \( n^{th} \) root of unity in some extension field of \( F_q \), then \( g(x) = \prod_s M_{\alpha^s}(x) \), where the product is over a subset of representatives of the \( q \)-cyclotomic cosets modulo \( n \).

**Proof.** (1)&(2) Let \( g(x) \) be the monic minimal degree polynomial in \( C \). If \( c(x) \in C \), then \( c(x) = g(x)h(x) + r(x) \) and so \( r(x) \in C \). Thus, \( r(x) = 0 \), since the degree of \( r \) is less than that of \( g(x) \). This implies that \( \mathcal{C} = \langle g(x) \rangle \), which gives (1) and (2).

(3) By the Division Algorithm \( x^n - 1 = g(x)h(x) + r(x) \), with \( \deg(r(x)) < \deg(g(x)) \) or \( r(x) = 0 \). Since \( x^n - 1 \) corresponds to the zero codeword in \( C \), we know that \( r(x) \) must be in \( C \), which is a contradiction unless \( r(x) = 0 \). Hence, \( g(x)|x^n - 1 \).

(4)&(5) From (2), we know that
\[
\mathcal{C} = \langle g(x) \rangle = \{ g(x)h(x) : \deg(g) + \deg(h) < n \text{ or } h(x) = 0 \}
= \{ g(x)h(x) : \deg(h) < K \text{ or } h(x) = 0 \}.
\]
Thus, \( \dim \mathcal{C} \leq K \) and \( \{ g, xg, \ldots, x^{K-1}g \} \) all have different degrees and so are linearly independent, and so is a basis and the degree of \( \mathcal{C} \) is \( K \).

(6) Write out coefficients of the basis.

(7) Theorem 4.1.1 implies that \( x^{n-1} = \prod_s M_{\alpha^s}(x) \), where \( s \) ranges over representatives of \( q \)-cyclotomic cosets modulo \( n \). Three implies that \( g(x)|x^n - 1 \) and so \( g(x) = \prod_s M_{\alpha^s}(x) \) for \( s \) ranging over some subset of representatives of \( q \)-cyclotomic cosets modulo \( n \).

\[\square\]

**Corollary 4.2.2.** Let \( \mathcal{C} \neq 0 \) be cyclic and a subset of \( \mathcal{R}_n \). The following are equivalent:

(1) \( g(x) \) is the monic polynomial of minimal degree in \( \mathcal{C} \)

(2) \( \mathcal{C} = \langle g(x) \rangle \) where \( g(x) \) is monic and \( g(x)|x^n - 1 \).

**Definition 4.2.3.** We call such a \( g(x) \) the generator polynomial of \( \mathcal{C} \).

Note that there is a 1-1 correspondence between nonzero cyclic codes and divisors of \( x^n - 1 \) that are not actually \( x^n - 1 \).

**Corollary 4.2.4.** The number of cyclic codes in \( \mathcal{R}_n \) is \( 2^m \), where \( m \) is the number of \( q \)-cyclotomic cosets modulo \( n \). Moreover, \( \dim \mathcal{C} \) is all possible sums of sizes of the \( q \)-cyclotomic cosets modulo \( n \).

**Corollary 4.2.5.** Let \( \mathcal{C}_1 \) and \( \mathcal{C}_2 \) be cyclic codes over \( F_q \) with generator polynomials \( g_1(x) \) and \( g_2(x) \), respectively. Then \( \mathcal{C}_1 \subseteq \mathcal{C}_2 \) if and only if \( g_2(x)|g_1(x) \).

**Theorem 4.2.6.** The dual of a cyclic code is cyclic.
Theorem 4.2.7. Let \( C \) be an \([n,k]\) cyclic code with generator polynomial \( g(x) \). Let \( h(x) = \frac{x^{n-1}}{g(x)} = \sum_{i=0}^{K} h_i x^i \). Then the generator polynomial of \( C^\perp \) is

\[
g^\perp(x) := \frac{x^k h(x^{-1})}{h(0)}.
\]

Also a generator matrix for \( C^\perp \) including a parity check matrix for \( C \) given by

\[
\begin{bmatrix}
h_k & h_{k-1} & \cdots & h_0 & 0 \\
h_k & h_{k-1} & \cdots & h_0 & \ddots \\
0 & h_k & h_{k-1} & \cdots & h_0
\end{bmatrix}
\]

There are three ways of encoding cyclic codes of length \( n \) over \( \mathbb{F}_q \) with generator polynomial \( g(x) \) of degree \( n - k \) so that \( C \) has dimension \( k \).

1. Let \( G \) be the generator matrix from the shifts of \( g(x) \) in Theorem 4.2.1. Encode the message \( m \in \mathbb{F}_q^n \) with codeword \( c = mG \).
2. The polynomial \( m(x) \) is associated to the message \( m \) with the largest degree being \( k - 1 \) (or the 0 polynomial). The polynomial \( x^{n-k} m(x) \) has the largest degree being \( n - 1 \) and its first \( n - k \) coefficients equaling 0. Therefore, the message is contained in the coefficients of \( x^{n-k}, x^{n-k+1}, \ldots, x^{n-1} \). Then, following the division algorithm, we have

\[
x^{n-k} m(x) = q(x) a(x) + r(x),
\]

where \( \text{deg}(r(x)) < n - k \) or \( r(x) = 0 \). Let \( c(x) = x^{n-k} m(x) - r(x) = q(x) a(x) \). So \( c(x) \in C \). Note that \( c(x) \) is not the same as \( x^{n-k} m(x) \) in the coefficients of \( 1, x, \ldots, x^{n-k-1} \) as the degree of \( r(x) \) is less than \( n - k \). Then \( c(x) \) holds the message \( m \) in the coefficients with terms of the degree at least \( n - k \).
3. Let \( C = \langle g(x) \rangle \). Then using the generator polynomial \( g^\perp(x) \) of the dual code \( C^\perp \) from Theorem 4.2.7, \( C \) is an \([n,k]\) code if \( c_0, c_1, \ldots, c_{n-k-1} \in C \) where \( c_0, \ldots, c_{n-1} \) from \( Hc^\top = 0 \). Scale the rows of \( H \) so that its row shift on the monic polynomial \( g^\perp(x) = h_0' + h_1' x + \cdots + h_{k-1}' x^{k-1} + x^k \). To encode \( c \), choose \( k \) information bits \( c_0, c_1, \ldots, c_{k-1} \) so that \( c_i = \sum_{j=0}^{k-1} h_j' c_{i-k+j} \) is performed in the order \( i = k, k+1, \ldots, n-1 \).

Each of these procedures can be implemented using linear shift registers. These are series of delay elements where one bit enters one end of the shift register and moves to the delay element with each new clock cycle. A binary linear feedback shift register is a linear shift register where the output is fed back into the shift register as part of the input. See Figure 4.2 and Example 4.2.10 on page 130 in the text for an example of an implemented circuit.

Definition 4.2.8. A code \( C \) that has the property that there exists an \( s \in \mathbb{Z} \) so that the shift of a codeword by \( s \) positions is again a codeword is called a quasi-cycle code.

Definition 4.2.9. A double circulant code is a code \( C \) where the generator matrix is \([I,A]\) with \( A \) a circulant matrix.
4.3. Idempotents and multipliers. Monday, June 27, 2005:

**Definition 4.3.1.** Let \( R \) be a ring. Then \( e \in R \) is an **idempotent** if \( e^2 = e \). A **unity** in \( R \) is a nonzero multiplicative identity.

**Theorem 4.3.2.** Let \( C \) be a cyclic code in \( R_n \). Then:

(i) There exists an idempotent \( e(x) \in C \) such that \( C = \langle e(x) \rangle \)

(ii) If \( e(x) \) is a nonzero idempotent in \( C \), then \( C = \langle e(x) \rangle \) if and only if \( e(x) \) is a unity of \( R \).

**Proof.** If \( C = \{0\} \), let \( e(x) = 0 \). Then (i) is clear. Suppose then that \( C \neq \{0\} \).

Suppose \( e(x) \) is a unity in \( C \); then \( \langle e(x) \rangle \subset C \). If \( c(x) \in C \), then \( c(x)e(x) = c(x) \). Therefore, \( c(x) \notin \langle e(x) \rangle \), and \( \langle e(x) \rangle = C \).

Suppose \( e(x) \neq 0 \), then there is a idempotent such that \( C = \langle e(x) \rangle \). If \( c(x) \in C \), then \( c(x) = f(x)e(x) \). But, \( c(x)e(x) = f(x)e(x)^2 = f(x)e(x) = c(x) \). Therefore, \( e(x) \) is a unity.

For part (i), suppose \( C = \langle e_1(x) \rangle = \langle e_2(x) \rangle \), where both \( e_i \) are idempotents. By (ii), \( e_i(x) \) are both units. Because units are unique, \( e_1(x) = e_2(x) \).

If \( g(x) \) is the generating polynomial \( C \), then \( g(x) | x^n - 1 \). If

\[
h(x) = \frac{x^n - 1}{g(x)}
\]

then \( \gcd(g, h) = 1 \). So, there exists \( a(x), b(x) \in \mathbb{F}(x) \) such that \( 1 = a(x)g(x) + b(x)h(x) \). Let \( e(x) := a(x)g(x) \mod x^n - 1 \). Then,

\[
e(x)^2 = (a(x)g(x))(1 - b(x)h(x))
= a(x)g(x) - a(x)b(x)g(x)h(x)
= a(x)g(x)(\mod x^n - 1)
= e(x)
\]

Hence, \( e(x) \) is an idempotent. Also, if \( c(x) \in C \), then \( c(x) = f(x)g(x) \). Therefore,

\[
c(x)e(x) = f(x)g(x)(1 - b(x)h(x)) = f(x)g(x)(\mod x^n - 1)
= c(x)
\]

Hence, \( e(x) \) is unity. By part (ii), \( C = \langle e(x) \rangle \). \( \square \)

**Theorem 4.3.3.** Let \( C \) be a cyclic code over \( \mathbb{F}_q \) with the generating idempotent \( e(x) \). Then the generating polynomial of \( C \) is

\[
g(x) = \gcd(e(x), x^n - 1)
\]

in \( \mathbb{F}_q[x] \).

**Proof.** Let \( d(x) = \gcd(e(x), x^n - 1) \), and let \( g(x) \) be the generating polynomial for \( C \). Then, \( d(x) | e(x) \) implies that \( d(x)k(x) = e(x) \) and \( \langle e(x) \rangle = C \subset \langle d(x) \rangle \). By theorem 4.2.1, \( g(x) | x^n - 1 \) and \( g(x) | e(x) \). Therefore, \( g(x) | d(x) \) and \( C = \langle g(x) \rangle \supset \langle d(x) \rangle \).

Thus, \( C = \langle d(x) \rangle \).

But \( d(x) \) is monic, divides \( x^n - 1 \), and generates \( C \) so corollary 4.2.2 implies that \( g(x) = d(x) \). \( \square \)
Example: 4.3.4 from the book. In $\mathbb{F}_2$, 
\[ x^7 - 1 = (1 + x)(1 + x + x^3)(1 + x^2 + x^3) \]
So, there exists $2^7$ generating polynomials, and thus 8 codes of length 7 over $\mathbb{F}_2$.

**Theorem 4.3.4.** Let $C = [n, k]$ cyclic code with generating idempotent 
\[ e(x) = e_0 + \cdots + e_{n-1}x^{n-1} \]
Then, 
\[
\begin{bmatrix}
  e_0 & e_1 & \cdots & e_{n-2} & e_{n-1} \\
  e_{n-1} & e_0 & \cdots & e_{n-3} & e_{n-2} \\
  \vdots & & & & \\
  e_{n-k+1} & e_{n-k+2} & \cdots & e_{n-k-1} & e_{n-k}
\end{bmatrix}
\]

**Proof.** It is enough to show that $\{e(x), xe(x), \ldots, x^{k-1}e(x)\}$ is a basis of $C$. This can be simplified to showing that if $a(x) \in \mathbb{F}_q[x]$ with $\deg(a(x)) \leq k - 1$ and $a(x)e(x) = 0$, then $a(x) = 0$. If $a(x)e(x) = 0$, then $a(x)e(x)g(x) = 0$. This implies that $a(x)g(x) = 0$ and $a(x) = 0$, by Theorem 4.2.1 v). $\square$

**Definition 4.3.5.** 
$C_1 + C = \{e_1 + e_2 : e_1 \in C_1, e_2 \in C_2\}$

**Theorem 4.3.6.** Let $C_i$ be cyclic codes over $\mathbb{F}_q$ with generating polynomials $g_i(x)$ and generating idempotents $e_i(x)$. Then,

(i) $C_1 \cap C_2$ has $g(x) = \gcd(g_1(x), g_2(x))$ and $e(x) = e_1(x)e_2(x)$.

(ii) $C_1 + C_2$ has $g(x) = \lcm(g_1(x), g_2(x))$ and $e(x) = e_1(x) + e_2(x) - e_1(x)e_2(x)$.

**Definition 4.3.7.** I (an ideal of $\mathcal{R}$) is minimal provided there is no proper ideal between $\{0\}$ and $I$.

**Notation:**
Let $x^n - 1 = f_1(x) \cdots f_s(x)$, where $f_i$ is irreducible over $\mathbb{F}_q$ for $1 \leq i \leq s$. Let 
\[ \hat{f}_i(x) = \frac{x^n - 1}{f_i(x)} \]
Let $\hat{e}_i(x)$ be the generating idempotent of $\langle \hat{f}_i(x) \rangle$.

**Definition 4.3.8.** $\{\hat{e}_i(x)\}$ are called the primitive idempotents of $\mathcal{R}_n$.

**Theorem 4.3.9.** We use the notation from above.

(i) $\langle \hat{f}_i(x) \rangle$ are the minimal ideals in $\mathcal{R}_n$.

(ii) $\mathcal{R}_n = \sum_1^s \langle \hat{f}_i \rangle$ as a vector space

(iii) $i \neq j$ implies that $\hat{e}_i(x)\hat{e}_j(x) = 0$ in $\mathcal{R}_n$

(iv) $\sum_1^s \hat{e}_i = 1$ in $\mathcal{R}_n$.

(v) The only idempotents in $\langle \hat{f}_i \rangle$ are 0 and $\hat{e}_i$

(vi) If $e(x) \neq 0$ is an idempotent in $\mathcal{R}_n$, then there exists a subset $T$ of $\{1, \ldots, s\}$ such that $e(x) = \sum_{i \in T} \hat{e}_i(x)$ and $\langle e(x) \rangle = \sum_{i \in T} \langle \hat{f}_i(x) \rangle$

**Note:** this theorem allows us to produce all idempotents in $\mathcal{R}_n$ and hence all cyclic codes, once the primitive idempotents are known.
Theorem 4.3.10. Let $M$ be a minimal ideal of $\mathcal{R}_n$. Then $M$ is an extension field of $\mathbb{F}_q$.

Proof. It is enough to show that $M \subset \mathbb{F}_q[x]$ is a field. Let $a \in M \setminus \{0\}$. Then $0 \not\subseteq \langle a \rangle \subset M$. By the minimality of $M$, we know that $\langle a \rangle = M$. Let $e$ be the unity in $M$. Then, there exists $b \in \mathcal{R}_n$ so that $ab = e$. Set $c = be \in M$, then $ac = abc = e^2 = e$. Thus $a$ is a unit.

Theorem 4.3.11. Let $C_i$ be a cyclic code of length $n$ over $\mathbb{F}_q$, for $1 \leq i \leq a$. Then,

$$
dim(C_1 + \cdots + C_a) = \sum_{i=1}^{a} \dim(C_i) - \sum_{1 \leq i < j \leq a} \dim(C_i \cap C_j)
$$

$$
+ \sum_{1 \leq i < j \leq k \leq a} \dim(C_i \cap C_j \cap C_k) - \cdots + (-1)^{a-1} \dim(C_1 \cap \cdots \cap C_a)
$$

Definition 4.3.12. For $a \in \mathbb{Z}$ with $\gcd(a, n) = 1$, then the function $\mu_a$ given by $i \cdot \mu_a \equiv a \ (\text{mod } n)$ is called a multiplier.

Remarks:

1. $\mu_a$ is a permutation of $\mathbb{Z}/\mathbb{Z}_n$
2. $\mu_a$ acts on $\mathcal{R}_n$ by $f(x)\mu_a \equiv f(x^a)(\text{mod } x^n - 1)$.

Theorem 4.3.13. Let $f(x), g(x) \in \mathcal{R}_n$ and suppose $e(x)$ is an idempotent of $\mathcal{R}_n$ and $\gcd(a, n) = 1$. Then:

(i) if $b \equiv_n a$, then $\mu_b = \mu_a$.
(ii) $(f(x) + g(x))\mu_a = f(x)\mu_a + g(x)\mu_a$
(iii) $(f(x)g(x))\mu_a = (f(x)\mu_a)(g(x)\mu_a)$
(iv) $\mu_a$ is an automorphism of $\mathcal{R}_n$.
(v) $e(x)\mu_a$ is an idempotent of $\mathcal{R}_n$.
(vi) $\mu_a$ leaves invariant each $q$-cyclotomic coset modulo $n$ and has order equal to $\text{ord}_n(q)$.

Theorem 4.3.14. Let $C$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with generating idempotent $e(x)$. Let $a$ be an integer with $\gcd(a, n) = 1$. Then,

(i) $C_{\mu_a} = \langle e(x)\mu_a \rangle$ and $e(x)\mu_a$ is the generating idempotent of the cyclic code $C_{\mu_a}$
(ii) $e(x)\mu_a = e(x)$ and $\mu_a \in \text{PAut}(C)$.

Corollary 4.3.15. Let $C$ be a cyclic code of length $n$ over $\mathbb{F}_q$. Let $A$ be the group of order $n$ generated by the cyclic shift $i \mapsto i + 1 \mod n$. Let $B$ be the group of order $\text{ord}_n(q)$ generated by $\mu_q$. Then the group $G$ of order $n$ (with $\text{ord}_n(q)$) generated by $A$ and $B$ is a subgroup of $\text{PAut}(C)$.

Corollary 4.3.16. Let $C$ be a cyclic code of length $n$ over $\mathbb{F}_q$ with generating idempotent $e(x) = \sum_{i=1}^{n-1} e_i x^i$. Then,

(i) $e_i = e_j$ if $i$ and $j$ are in the same $q$-cyclotomic coset modulo $n$.
(ii) If $q = 2$, then

$$
e(x) = \sum_{j \in J} \sum_{i \in C_j} x
$$

where $J$ is a set of coset representative of the $q$-cyclotomic cosets modulo $n$.
(iii) If \( q = 2 \), and \( e(x) \) has the form given in the above formula, then \( e(x) \) is an idempotent in \( \mathbb{R}_n \).

**Theorem 4.3.17.** Let \( C_1 \) and \( C_2 \) be cyclic codes of length \( n \) over \( \mathbb{F}_q \) and assume that \( \gcd(n, \phi(n)) = 1 \). Then \( C_1 \) and \( C_2 \) are permutation equivalent codes if and only if there is a multiplier \( \mu_n \) that maps \( C_1 \) to \( C_2 \).

4.4. **Zeros of Cyclic Codes.** This will allow us to construct a parity check matrix for cyclic codes.

Let \( t = \text{ord}_n(q) \). Fix a primitive \( n \)-th root of unity of \( \alpha \in \mathbb{F}_{q^t} \). Let \( C = \langle g(x) \rangle \).

By Theorem 4.2.1 vii,

\[ g(x) = \prod_s M_{\alpha^s}(x) \]

Where \( s \) goes through a subset of the representatives of \( q \)-cyclotomic cosets modulo \( n \). By Theorem 4.1.1,

\[ g(x) = \prod_s \prod_{i \in C_s} (x - \alpha^i) \]

Let

\[ T = \bigcup_s C_s \]

so

\[ g(x) = \prod_{i \in T} (x - \alpha^i). \]

**Definition 4.4.1.**

\[ Z = \{ \alpha^i : i \in T \} \]

is the set of **zeros** of \( C \). \( \{ \alpha^i : i \notin T \} \) is the set of **nonzeros** of \( C \). \( T \) is the **defining set** of \( C \).

Since \( c(x) \in C \) if and only if \( g(x) | c(x), c(x) \in C \Leftrightarrow c(\alpha^i) = 0 \forall i \in T \). Theorem 4.2.1 gives \( \dim C = n - \deg g(x) = n - |T| \).

**Theorem 4.4.2.** We have the following:

1. \( T \) is a union of \( q \)-cyclotomic cosets
2. \( g(x) = \prod_{i \in T} (x - \alpha^i) \)
3. \( c \in \mathbb{R}_n \) is in \( C \) if and only if \( c(\alpha^i) = 0 \) for all \( i \in T \).
4. \( \dim C = n - |T| \).

Let \( T = C_{i_1} \cup \cdots \cup C_{i_w} \). By 3, \( c(x) \in C \) if and only if \( c(\alpha^{i_1}) = \cdots = c(\alpha^{i_w}) = 0 \).

Also,

\[ C_{i_1} = \{ i_1, qi_1, \ldots, q^{w-1}i_1 \} \]

So,

\[ c(\alpha^{q^{i_1}}) = c(\alpha^{i_1})^q \]

We have \( c(\alpha^{i_1}) = 0 \) if and only if

\[
\begin{bmatrix}
1, \alpha^{i_1}, \ldots, \alpha^{(n-1)i_1}
\end{bmatrix}
\begin{bmatrix}
c_0 \\
c_1 \\
\vdots \\
c_{n-1}
\end{bmatrix} = 0
\]
Again, \( c \in C \) if and only if \( Lc^t = 0 \), where

\[
L = \begin{bmatrix}
1 & \alpha^{i_1} & \ldots & \alpha^{(n-1)i_1} \\
1 & \alpha^{i_2} & \ldots & \alpha^{(n-1)i_2} \\
& & \ddots & \vdots \\
1 & \alpha^{i_n} & \ldots & \alpha^{(n-1)i_n}
\end{bmatrix}
\]

Because \( \alpha^i \) may not be in \( \mathbb{F}_q \), we can pick an appropriate basis of \( \mathbb{F}_q^t \) as a vector space over \( \mathbb{F}_q \). Hence, we can get a \( H \in M_{tw \times n}(\mathbb{F}_q) \) such that \( Hc^T = 0 \) if and only if \( c \in C \). In order to assure that \( H \) is a parity check matrix, we may have to eliminate rows to make them all linearly independent.

We can discover other generators for \( C = \langle g(x) \rangle \):

**Theorem 4.4.3.**

\( C = \langle v(x) \rangle \iff \gcd(v(x), x^n - 1) = g(x) \)

**Proof.** Suppose \( C = \langle v(x) \rangle \), let \( d(x) = \gcd(v(x), x^n - 1) \). We have \( g|\text{r} \) and \( g|x^n - 1 \) so \( g|d \). Because there exists a \( a(x) \in \mathbb{R}_n \) such that \( g(x) = v(x)a(x) \), there exists a \( b(x) \) in our original ring such that \( g(x) = v(x)a(x) + b(x)(x^n - 1) \). Hence \( d(x)|g(x) \), and \( g(x) = d(x) \).

Suppose \( \gcd(v(x), x^n - 1) = g(x) \). So, \( g(x)|v(x) \) and \( \langle v(x) \rangle \subset \langle g(x) \rangle = C \). By the Euclidean Algorithm,

\[ g(x) = a(x)v(x) + b(x)(x^n - 1) \]

By modding out,

\[ g(x) = a(x)v(x) \]

so \( \langle g(x) \rangle \subset \langle v(x) \rangle \), completing the proof. \( \square \)

As a corollary, we have the following:

**Corollary 4.4.4.** Let \( C \) be a cyclic code of length \( n \) with defining set \( T \). Let \( \alpha \) be a primitive \( n \)-th root of unity and \( a \in \mathbb{Z} \) such that \( \gcd(n, a) = 1 \). Then \( \{\alpha^{a^{-1}i} : i \in T\} \) is the zeros \( C_{\mu_a} \) and \( \alpha^{-1}T \mod n \) is the defining set.

**Complements of a code:**

**Definition 4.4.5.** Let \( C \) be any code of length \( n \). A complement \( C^c \) is a code such that \( C \oplus C^c = \mathbb{F}_q^n \).

**Theorem 4.4.6.** 4.4.6 in the book:

Let \( C \) be a cyclic code with generator \( g(x) \), generating idempotent \( e(x) \) and defining set \( T \), then

\[ h(x) = \frac{x^n - 1}{g(x)} \]

generates \( C^c \),

\[ 1 - e(x) \]

is the generating idempotent, and

\( \{0, 1, \ldots, n - 1\} \setminus T \)

is the defining set.
**Definition 4.4.7.** The reciprocal polynomial of
\[ f(x) = f_0 + f_1 x + \cdots + f_a x^a \]
is
\[ f^*(x) = x^a f(x^{-1}) = f_a + f_{a-1} x + \cdots + f_0 x^a \]

**Remark:**
\[(f_1 f_2)^* = f_1^* f_2^* \]
yet
\[(f_1 + f_2)^* \neq f_1^* + f_2^* \]

**Definition 4.4.8.** We say that \( f(x) \) is reversible if \( f(x) = f^*(x) \).

**Lemma 4.4.9.** Let \( f(x) \in \mathbb{F}_q[x] \).

(i) If \( \beta_1, \ldots, \beta_r \) are non-zero roots of \( f \) in some extension of \( \mathbb{F}_q \), then \( \beta_1^{-1}, \ldots, \beta_r^{-1} \) are non-zero roots of \( f^* \).

(ii) If \( f(x) \) is irreducible in \( \mathbb{F}_q[x] \), so too is \( f^* \).

(iii) If \( f(x) \) is primitive, then so too is \( f^* \).

**Lemma 4.4.10.** Let \( a = a_0 a_1 \cdots a_{n-1}, b = b_0 \cdots b_{n-1} \) in \( \mathbb{F}_q^n \) with associated polynomial \( a(x), b(x) \). Then \( a \perp b \) (and all shifts of \( b \)) in \( \mathbb{F}_2^n \) iff \( a(x) b^*(x) = 0 \) in \( \mathcal{R}_n \).

**Theorem 4.4.11.** Recall that if \( C \) is a code over \( \mathbb{F}_4 \), then \( C^\perp \perp = (\overline{C})^\perp \). Let \( C \) be an \([n, k]_c \) cyclic code over \( \mathbb{F}_q (\mathbb{F}_4) \) with generating polynomial \( g(x) \), generating idempotent \( e(x) \) (\( e'(x) \)), defining set \( T \). Let
\[ h(x) = \frac{x^n - 1}{g(x)} \]
(this is called the check polynomial for \( C \)). Then,

1. \( C^\perp \) is cyclic (\( C^\perp \perp \) is cyclic) and \( C^\perp = C^\perp \mu_{-1} \) (\( C^\perp \perp = C^\perp \mu_{-2} \))

2. \( C^\perp \) has generating idempotent \( 1 - e(x) \mu_{-1} \) (\( 1 - e(x) \mu_{-2} \)) and generating polynomial
\[ h^*(x) := \frac{x^k}{h(0)} h(x^{-1}) \]

**Corollary 4.4.12.** Let \( C \) be a cyclic code over \( \mathbb{F}_q \) of length \( n \), \( g(x) \) the generator, and define
\[ h(x) = \frac{x^n - 1}{g(x)} \]
Then, \( C \) is self-orthogonal if and only if \( h^*(x) \mid g(x) \).