Cyclic Error-Correcting Codes

Daniel F. Russell
December 14, 2001
Dr. Edward Green, Advisor
Information is passed every day in our society. It is essential that interference in the communication of this information hinders the information from being received as little as possible. Error-correcting codes provide us with this ability. Error-correcting codes allow us to receive a piece of information, identify any errors, locate them, and correct them. Cyclic codes are an especially useful kind of error-correcting code, and BCH codes and QR codes are especially useful kinds of cyclic codes. Error-correcting code theory has also been used in areas outside of information communication. Error-correcting code theory is an important subject to study.

We will restrict our study to linear codes, that is, if \( a \) and \( b \) are codewords in a code \( C \), then any linear combination of \( a \) and \( b \) are codewords in \( C \). Consider the vector space \( V \) of all \( n \)-tuples of 0’s and 1’s, with addition component-wise mod 2. An \([n,k]\) code is the set of all linear combinations of \( k \) independent vectors in \( V \). So an \([n,k]\) binary code has \( 2^k \) codewords in it. This code has dimension \( k \), and can be given by a basis. The generator matrix has the basis vectors of the code as the horizontal rows. For example, consider the \([7, 4]\) binary code \( C_1 \) whose generator matrix is

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

All codewords of \( C_1 \) are linear combinations of these 4 vectors. \( C_1 \) has \( 2^4 = 16 \) codewords. The information set of \( C_1 \) is any independent set of \( k = 4 \) columns in \( G_1 \). Say that the information set is the first 4 columns (so the 5th, 6th and 7th columns are redundancy positions).
Another way to represent codes in the parity check matrix. The rows of a parity check matrix are made up of the equations for the redundancy positions (the parity check equations). In our example, we can compute the redundancy positions by the equations:

\[
\begin{align*}
a_5 &= a_1 + a_3 \\
a_6 &= a_1 + a_2 + a_3 \\
a_7 &= a_4.
\end{align*}
\]

These equations were found by examining the generator matrix, but they will hold for any codeword in \( C_1 \). The parity check matrix for \( C_1 \) is then:

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

This is because a vector \((a_1, a_2, a_3, a_4, a_5, a_6, a_7)\) is orthogonal to the first row of \( H_1 \) if \( a_1 + a_3 + a_5 = 0 \), which is the same as \( a_5 = a_1 + a_3 \) (in binary), which is our first parity check equation. The other two rows of \( H_1 \) relate to the other two parity check equations similarly. So \( C_1 \) is the set of all 7-tuples that are orthogonal to each row of \( H_1 \).

The weight of a codeword is the number of its nonzero digits. The weight of the first row of \( G_1 \) is 3. The minimum weight of a code, \( d \), is the weight of the nonzero vector of smallest weight in the code. The minimum weight of a code can be found by examining the combinations of the rows of the generator matrix. The row of \( G_1 \) of smallest weight has a weight of 2. Any linear combination of rows will have a weight greater than 2 since each row has exactly one 1 in the first 4 digits, all in different positions, and a row added to itself would result in the zero vector. A code is often expressed as \([n,k,d]\), where \( d \) is the minimum weight. So \( C_1 \) is a \([7,4,2]\) code.

The distance between two vectors \( u, v \) is the number of positions in which they differ, denoted \( d(u,v) \). The distance between the first 2 rows of \( G_1 \) is 3. Distance is important in decoding. If a vector \( v \) is received, one good strategy for decoding it would
be to find the codeword \( u \) in the code such that \( d(u-v) \) is minimized. This is called maximum likelihood decoding. We call \( u - v \) the error vector. If \( d \) is the minimum weight of a code \( C \), then \( C \) can correct \( t = [\frac{(d-1)}{2}] \) or fewer errors. So \( C_1 \) can decode \([2-1] / 2 = 0 \) errors. So \( C_1 \) is an ineffectual code. The Hamming \([7,4]\) code, though, has a generator matrix

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

so it has a minimum weight of 3, and can correct \([3-1] / 2 = 1 \) error.

One special kind of code is a cyclic code. Cyclic codes correspond to ideals in certain commutative rings. A commutative ring \( R \) with unit is much like a field, but a nonzero element need not have an inverse. An ideal \( I \) in a commutative ring \( R \) is a set of elements in \( R \) such that (1) If \( a \) is in \( I \), then \( ab \) is in \( I \) for all \( b \) in \( R \) and (2) If \( a \) and \( b \) are in \( I \), then \( a + b \) and \( a - b \) are in \( I \). Let \( R = F[x]/(x^n - 1) \) where \( F \) is a field. A cyclic code \( C \) corresponds to an ideal in \( R \). An ideal \( I \) in \( R \) is called a principal ideal if every element of \( I \) is a multiple of a fixed polynomial \( g(x) \). If \( C \) corresponds to the ideal in \( R = F[x]/(x^n - 1) \), then \( g(x) \) divides \((x^n - 1)\). If \( g(x) = g_0 + g_1x + g_2x^2 + \ldots + g_{n-k}x_{n-k} \), then a generator matrix of \( C \) is:

\[
\begin{bmatrix}
g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\
g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\
g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 0 & g_{n-k+1} & g_{n-k} \\
0 & 0 & 0 & \cdots & 0 & g_{n-k+1} & g_{n-k} \\
\end{bmatrix}
\]
Note $G$ has rank $k$. For an example, we will construct the generator matrix for the cyclic code for $n = 7$ over the Reals. First, we factor $x^7 - 1 = (x + 1) (1 + x + x^3) (1 + x^2 + x^3)$. Choose the generator polynomial $g(x)$ to be $1 + x^2 + x^3$. So $C_3 = \langle g(x) \rangle$ is 4 dimensional and has the generator matrix:

$$
G_3 = \begin{pmatrix}
1 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 & 1
\end{pmatrix}.
$$

To find the parity check matrix, we find $h(x)$ such that $x^n - 1 = g(x)h(x)$. $h(x)$ is called the check polynomial. If $h(x) = h_0 + h_1x + h_2x^2 + \ldots + h_kx^k$, then the parity check matrix $H$ of $C$ is:

$$
H = \begin{pmatrix}
h_k & h_{k-1} & h_{k-2} & \ldots & h_0 & 0 & 0 & \ldots & 0 & 0 \\
0 & h_k & h_{k-1} & \ldots & h_2 & h_0 & 0 & \ldots & 0 & 0 \\
0 & 0 & h_k & \ldots & 0 & 0 & 0 & \ldots & 0 & 0
\end{pmatrix}.
$$

The reciprocal polynomial of $h(x)$ can be used to construct the parity check matrix the same way $g(x)$ was used for the generator matrix. So for our $n = 7$ example, $h(x) = (1 + x)(1 + x + x^3) = 1 + x^2 + x^3 + x^4$. The reciprocal polynomial then is $x^4(1 + x^2 + x^3 + x^4) = 1 + x + x^2 + x^4$. So the parity check matrix is

$$
H_3 = \begin{pmatrix}
1 & 1 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 1
\end{pmatrix}.
$$

The generator polynomial gives us much information about a code. Unfortunately, it is not always easy to find, as $x^n - 1$ is often quite difficult to factor.
Idempotent generators, though, can be found without factoring $x^n - 1$. An idempotent is a polynomial $a(x)$ such that $a^2(x) = a(x)$. A generator $e(x)$ of an ideal in $R$ is called an idempotent generator if it is an idempotent. Every cyclic code has an idempotent generator. The matrix formed by $e(x)$ and its next $k-1$ cyclic shifts form a generator matrix for the code.

Idempotent generators are found from the union of cyclotomic cosets of $n$. A cyclotomic coset $C_i$ is found by

$$C_i = i \cdot p^m \pmod{n}$$

for all $m$ (for binary codes, $p$ is 2). The idempotent generators $f(x)$ are formed by letting unions of the cyclotomic cosets $C_i$, for all $i$ such that $0 \leq i \leq n$, be powers of $x$ that occur in $f(x)$ with nonzero coefficients.

For example, we find the idempotent generators for $n=7$. Letting $p = 2$, the cyclotomic cosets of $n$ are $C_0 = \{0\}$, $C_1 = \{1, 2, 4,\}$, and $C_3 = \{3, 6, 5\}$. So the idempotents for $x^7 - 1$ are $0, 1, x + x^2 + x^4, x^3 + x^5 + x^6, 1 + x + x^2 + x^4, 1 + x^3 + x^5 + x^6, x + x^2 + x^3 + x^4 + x^5 + x^6$, and $1 + x + x^2 + x^3 + x^4 + x^5 + x^6$. All but 0 and 1 are idempotent generators of $x^7 - 1$, since 1 generates the whole space and 0 generates the zero vector.

Let $k$ be the number of cyclotomic cosets for $n$. Let $m$ be the smallest integer such that $2m \equiv 1 \pmod{p}$. Then $k - 1$ of the cyclotomic cosets have $m$ elements, and one has one element (zero). If $e(x)$ is the idempotent generator of a code $C$, then the generator polynomial $g(x)$ of $C$ equals the g.c.d.($e(x)$, $(x^n - 1)$). In our example, g.c.d.($(x + x^2 + x^4)$, $(x^7 - 1)$) = $1 + x + x^3$, so the generator polynomial of $x^7 - 1$ related to $x + x^2 + x^4$ is $1 + x + x^3$. 
Bose-Chaudhuri-Hocquenghem (BCH) codes are a special family of cyclic codes designed to correct a certain number of errors. BCH codes are defined in terms of the roots of their generator polynomials. BCH codes are defined in the following way: Let $C$ be a cyclic code of length $n$ over $GF(q)$ where $g.c.d.(n, q) = 1$. Let $\alpha$ be a primitive $n$th root of $GF(q^m)$, where $m$ is the order of $q$ mod $n$. $C$ is a BCH code of designed distance $\delta$ if the generator polynomial $g(x)$ of $C$ is the product of distinct minimal polynomials of the $\delta - 1$ elements $\alpha^b, \alpha^{b+1}, \ldots, \alpha^{b+\delta-2}$. The $\delta - 1$ consecutive roots are $\alpha, \alpha^2, \ldots, \alpha^{\delta - 1}$ if $b = 1$. $C$ is called a primitive BCH code if $n = q^m - 1$. The minimum weight if a BCH code of designed distance $\delta$ is at least $\delta$. Every vector in $C$ is orthogonal to each row of matrix $H$:

We do not know the dimension $k$ of our code $C$, though. The matrix $H$ has $\delta - 1$ rows over $GF(q^m)$, and thus $(\delta - 1)m$ rows over $GF(q)$. So $k \geq n - m(\delta - 1)$. This is not too helpful. It can be proved, though, that for any positive integers $m$ and $t \leq 2^{m-1} - 1$, there is a binary BCH code of length $n = 2^m - 1$ that is $t$-error correcting and has dimension $k \geq n - mt$. This is a much better bound on $k$, and is the reason why BCH codes are so practical.

Let us construct a binary double error-correcting BCH code of length 7, and call it $C$. So $q = 2$, $n = 7$, and $t = 2$. So $m$ is the order of 2 mod 7, or 3. Let $\alpha$ be a root of $x^3 + x + 1$, and thus a primitive root in $GF(8)$. To find $\delta$, the designed distance, we solve $2 = [(\delta-1) / 2]$. So $\delta = 5$. The generator polynomial $g(x)$ of $C$ must have roots $\alpha, \alpha^2, \alpha^3$, and
We know \( x^3 + x + 1 \) has root \( \alpha \), and thus \( \alpha^2 \) and \( \alpha^4 \). The inverse of \( x^3 + x + 1, x^3(x^{-1} + 1) = x^3 + x^2 + 1 \), then, has roots \( \alpha^{-1} = \alpha^6, \alpha^{-2} = \alpha^5, \) and \( \alpha^{-4} = \alpha^3 \). So let \( g(x) = (x^3 + x + 1)(x^3 + x^2 + 1) \), since this has roots \( \alpha, \alpha^2, \alpha^3, \) and \( \alpha^4 \) (among others). The parity check matrix of \( C \), then, is

\[
H = \begin{pmatrix}
1 & \alpha & \alpha^2 & \alpha^3 & \ldots & \alpha^6 \\
1 & \alpha^2 & \alpha^4 & \alpha^6 & \ldots & \alpha^8 \\
1 & \alpha^3 & \alpha^6 & \alpha^2 & \ldots & \alpha^9 \\
1 & \alpha^4 & \alpha & \alpha^5 & \ldots & \alpha^3 \\
\end{pmatrix}.
\]

Decoding BCH codes is an involved process. The following discussion will be for binary BCH codes with roots \( \alpha, \alpha^2, \ldots, \alpha^{2^t} \). For a received vector \( v \), Compute \( v(\alpha^i) = S_i \) where \( i = 1, 2, \ldots, 2t \). Next, determine the maximum number \( r \) such that the determinant of \( M \) is not 0, where

\[
M = \begin{pmatrix}
S_1 & S_2 & \ldots & S_x \\
S_2 & S_3 & \ldots & S_{x+1} \\
\vdots & \vdots & \ddots & \vdots \\
S_x & S_{x+1} & \ldots & S_{2^t-1} \\
\end{pmatrix}.
\]

Start with \( r = t \). If \( |M| = 0 \), let \( r = t-1 \), and so on until \( |M| \neq 0 \). Then \( r \) errors have occurred in \( v \). Now find the coefficients the error-locator polynomial,

\[
s(x) = 1 + s_1x + \ldots + s_x x^x = (1 - \alpha x_1)(1 - \alpha x_2) \ldots (1 - \alpha x_x),
\]

where \( x_j = \alpha^{j_{2^t}} \).

This can be done by finding coefficients of the equation

\[
S_{j+r} + s_1S_{j+r-1} + s_2S_{j+r-2} + \ldots + s_rS_j = 0 \quad \text{for } j = 1, \ldots, r.
\]

This can also be done using matrices by solving

\[
\begin{pmatrix}
-S_{x+1} \\
-S_{x+2} \\
\vdots \\
-S_{2^t}
\end{pmatrix}
\begin{pmatrix}
s_r \\
s_{x+1} \\
\vdots \\
s_{2^t}
\end{pmatrix}
= \begin{pmatrix}
s_x \\
s_{x+1} \\
\vdots \\
s_1
\end{pmatrix}
\]
where $M^{-1}$ is the inverse of the matrix $M$ given above. We can solve these equations because we know $S_i$ for $i \leq t$. Next finds the roots of $s(x)$, perhaps by just searching through the field. Finally, find the error locations and correct the errors.

Using our previous double error-correcting binary code example, assume we receive a vector $v(x) = x^7 + x^6 + x^4 + x^2 + 1$. To find $S_1$, we divide $v(x)$ by $x^3 + x + 1$ (since $\alpha$ is a root of $x^3 + x + 1$) to get a remainder of $x^2 + x + 1$. Since $\alpha^3 + \alpha + 1 = 0$, we know $\alpha + 1 = \alpha^3$, and since $\alpha^5 = \alpha^2 + \alpha^3$, we know $(\alpha^5)^2 + \alpha^5 + 1 = 0$. Thus a root of $x^2 + x + 1$ is $\alpha^5$ (Table 1 shows calculations of roots for all polynomials when $\alpha$ is a root of $x^3 + x + 1$).

Table 1: GF(8) where $\alpha$ is a root of the irreducible polynomial $f(x) = x^3 + x + 1$

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>$\alpha$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>100</td>
<td>$\alpha^2$</td>
<td>$\alpha^2$</td>
</tr>
<tr>
<td>011</td>
<td>$\alpha^3$</td>
<td>$\alpha$</td>
</tr>
<tr>
<td>110</td>
<td>$\alpha^4$</td>
<td>$\alpha^2$</td>
</tr>
<tr>
<td>111</td>
<td>$\alpha^5$</td>
<td>$\alpha^2$</td>
</tr>
<tr>
<td>101</td>
<td>$\alpha^6$</td>
<td>$\alpha^2$</td>
</tr>
</tbody>
</table>

So $S_1 = \alpha^5$. Then $S_2 = \alpha^3$ and $S_4 = \alpha^6$. To find $S_3$, divide $v(x)$ by $x^3 + x^2 + 1$ to get a remainder of $x^2 + 1$. Examining Table 1, we see that $\alpha^6$ is a root of $x^2 + 1$, and so $S_3 = \alpha^6$.

So

$$M = \begin{pmatrix} \alpha^5 & \alpha^3 \\ \alpha^3 & \alpha^6 \end{pmatrix} \text{ and } M^{-1} = \begin{pmatrix} \alpha^3 & 1 \\ 1 & \alpha^2 \end{pmatrix}.$$  

To find the coefficients of the error-locator polynomial, we find

$$\begin{pmatrix} \alpha^3 & 1 \\ 1 & \alpha^2 \end{pmatrix} \begin{pmatrix} \alpha^6 \\ \alpha^5 \end{pmatrix} = \begin{pmatrix} \alpha \\ \alpha^4 \end{pmatrix}.$$
So \( s(x) = 1 + \alpha x + \alpha^4 x^2 \). To find roots for \( s(x) \) we search through \( \text{GF}(8) \), and find \( \alpha^4 \) and \( \alpha^5 \). So the errors are at positions \( 7 - 4 = 3 \) and \( 7 - 5 = 2 \) (numbering the positions starting at 0 from right to left instead of left to right as in the beginning).

Another family of cyclic codes is quadratic residue (QR) codes. QR codes are defined over a finite field and easily constructed from their idempotent generators. For this discussion we will only consider binary QR codes. Let \( F \) be the field \( \text{GF}(p) \) where \( p \) is an odd prime, and let \( G \) be the multiplicative group of nonzero elements in \( F \). Note \( G \) has order \( p - 1 \). Let \( Q \) be a subgroup of \( G \) such \( q \in Q \) iff there exists a \( g \in G \) such that \( g^2 = q \). The elements of \( Q \) are called quadratic residues. \( Q \) has order \( (p-1)/2 \). It can be shown, through a difficult proof, that 2 is a quadratic residue iff \( p \equiv 1 \pmod{8} \) or \( p \equiv -1 \pmod{8} \). Also, -1 is a quadratic residue in \( \text{GF}(p) \) iff \( p \equiv 1 \pmod{4} \). For such a \( p \) let \( h \) be the subgroup of \( Q \) such that \( q \in H \) iff \( q = 2w \) for some \( w \). QR codes are sent onto themselves by permutations \( i \mapsto a_i \pmod{p} \) where \( a \in Q \). For any \( a \), if the permutation \( \mu_a \) defined by \( i \mu = a_i \pmod{n} \) sends a cyclic code \( C \) with idempotent \( e(x) \) onto a code \( C' \), then \( C' \) is cyclic with idempotent \( e(x) \mu_a \).

For the remainder of this discussion we will assume that \( p \equiv (+ or -)1 \pmod{8} \) so that 2 is a quadratic residue. The quadratic residues, then, are a union of cyclotomic cosets. Let \( m \) be the order of 2 mod \( p \). Then each nonzero cyclotomic coset has \( m \) elements. There are \( k \) of these cyclotomic cosets, where \( mk = p - 1 \). So the quadratic residues are a union of \( k/2 \) cyclotomic cosets, and nonresidues, \( N \), make up the union of the other \( k/2 \) cyclotomic cosets. The only \( e(x) \) that can be fixed by the quadratic residue permutation are

\[
1, \sum_{x \in \mathbb{Q}} x^i, \sum_{x \in \mathbb{Q}} x^{2i}, \sum_{i=0}^{\frac{p-1}{2}} x^i, 1 + \sum_{x \in \mathbb{Q}} x^i, 1 + \sum_{i=0}^{\frac{p-1}{2}} x^i, 1 + \sum_{i=0}^{\frac{p-1}{2}} x^i.
\]
We generally don’t consider \( 1, \sum_{i \in \mathbb{Q}^N} x^i, \) and \( 1 + \sum_{i \in \mathbb{Q}^N} x^i \) because their corresponding codes are \( R_p \), all even weight vectors, and \( <h> \) (the vector of all ones), respectively.

Let \( p \) be a prime such that \( p \equiv -1 (\text{mod } 8) \). Let \( e_1(x) = \sum_{i \in \mathbb{Q}^N} x^i \) and \( e_2(x) = \sum_{i \in \mathbb{N}^N} x^i \).

Let \( Q_1 \) be the QR code with idempotent \( e_1(x) \) and \( Q_2 \) be the QR code with idempotent \( e_2(x) \). Let \( Q'_1 \) be the QR code with idempotent \( 1 + e_2(x) \) and \( Q'_2 \) be the QR code with idempotent \( 1 + e_1(x) \). Then the four QR codes have the following properties:

(i) \( Q_1 \) and \( Q_2 \) are equivalent; \( Q'_1 \) and \( Q'_2 \) are equivalent.

(ii) \( Q_1 \cap Q_2 = <h> \), and \( Q_1 + Q_2 = R_p \).

(iii) \( \dim Q_1 = \dim Q_2 = (p + 1)/2 \).

(iv) \( Q_1 = Q'_1 + <h> \); \( Q_2 = Q'_2 + <h> \).

(v) \( \dim Q'_1 = \dim Q'_2 = (p - 1)/2 \).

(vi) \( Q'_1 \) and \( Q'_2 \) are self-orthogonal and \( Q_1^+ = Q'_1 \); \( Q_2^+ = Q'_2 \).

The situation when \( p \equiv 1 (\text{mod } 8) \) is similar. Rules (i) – (v) above apply if you change the definitions around so that \( Q_1 \) has the idempotent generator \( 1 + e_1(x) \), \( Q_2 \) has \( 1 + e_2(x) \), \( Q'_1 \) has \( e_2(x) \), and \( Q'_2 \) has \( e_1(x) \). Rule (vi) does not apply, and should be replaced with

(vi)' \( Q_1^+ = Q'_2 \) and \( Q_2^+ = Q'_1 \).

We can extend \( Q_1 \) and \( Q_2 \) by adding an overall parity check coordinate, labeled \( \infty \). The coordinates are now labeled \( \infty, 0, 1, \ldots, p - 1 \). I will call these new extended codes \( Q_{\infty 1} \) and \( Q_{\infty 2} \). \( Q_{\infty 1} \) and \( Q_{\infty 2} \) are equivalent \([p + 1, (p + 1)/2]\) codes. If \( p \equiv 1 (\text{mod } 8) \), \( Q_{\infty 1}^+ = Q_{\infty 2} \) and \( Q_{\infty 2}^+ = Q_{\infty 1} \). \( Q_{\infty 1} \) and \( Q_{\infty 2} \) contain only even weight vectors.

Define \( \text{PSL}_2(p) \) to be the group of permutations on \( \infty, 0, 1, \ldots, p - 1 \) for a prime \( p \) that is generated by the following permutations of elements \( i \) of \( \text{GF}(p) \):
\[ \sigma: \; i \rightarrow i + 1(\text{mod} p) \]

\[ \mu_a: \; i \rightarrow ai(\text{mod} p) \; \text{for} \; a \in Q \]

\[ \rho: \; i \rightarrow 1 / [i(\text{mod} p)]. \; i \neq 0. \]

Note that \( \sigma \) and \( \mu_a \) map \( \infty \) to \( \infty \) and \( \rho \) maps \( 0 \) to \( \infty \) and \( \infty \) to \( 0 \). It can be shown that \( \text{PSL}_2(p) \) has order \((p - 1)p(p + 1) / 2\). \( \text{PSL}_2(p) \) is contained in the group of \( \mathbb{Q}_{\infty 1} \) or of \( \mathbb{Q}_{\infty 2} \) (the group of a code is the set of all permutations that send \( C \) onto itself).

The minimum weight, \( d \), of a QR \([p, (p + 1)/2]\) code is odd. If \( p \equiv -1(\text{mod} \; 8) \), the number \( A_i \) of vectors of weight \( i \) in a QR code of length \( p \) is 0 unless \( i \equiv (0 \text{ or } 3)(\text{mod} \; 4) \).

If \( d \) is the minimum weight of a QR \([p, (p + 1)/2]\) code of length \( p \), then \( d^2 \geq p \). If \( p \equiv -1(\text{mod} \; 8) \), then \( d^2 - d + 1 \geq p \).

Now we will examine how to decode QR codes. Let \( y \) be the received vector of length \( p \). Let \( I \) be an information set. Let \( S = \{s_1, \ldots, s_r\} \) be the smallest set of permutations in \( G(C) \) chosen from combinations of \( \sigma, \mu_a, \) and \( \rho \) as defined above such that any given set of \( t \) coordinate positions is disjoint from at least one image of \( I \) under a permutation in \( S \). Let \( P \) be a set of parity check equations derived from idempotent generators of \( C \). Compute \( w_1 \) by encoding the positions \( I \) using \( P \). If \( d(w_1, y) \leq t \), decode \( y \) to \( w_1 \). Otherwise, compute \( P_2 = P\sigma_1 \), giving a new set of parity check equations. Next compute \( w_2 \) by encoding the information positions in \( I\sigma_1 \) with the parity checks in \( P_2 \). If \( d(w_2, y) \leq t \), decode \( y \) to \( w_2 \). Otherwise continue the above steps until either \( y \) is decoded or all the permutations in \( S \) have been used. It is possible that \( y \) will not be decoded.

Perhaps this decoding algorithm can best be explained through an example. We will use the \([7,4,3]\) QR code with idempotent \( e_1 = (0, 0, 0, 1, 0, 1, 1) \). We know \( 1 + e_2 = (1, 1, 1, 0, 1, 0, 0) \) and that its cyclic shifts generate the orthogonal code. We can assume
the first 4 positions \( \{0, 1, 2, 3\} \) are the information set. So we can derive the parity check equations:

\[
\begin{align*}
\alpha_4 &= \alpha_0 + \alpha_1 + \alpha_2, \\
\alpha_5 &= \alpha_0 + \alpha_1 + \alpha_3, \\
\alpha_6 &= \alpha_1 + \alpha_2 + \alpha_3.
\end{align*}
\]

Suppose that the sent vector \( x = e_1 \) and that the received vector \( y \) has an error in position 2, so \( y = (0, 1, 0, 1, 0, 1, 1) \). We compute \( w_1 \) by taking the same information set from \( y \) and using the parity check equations to find the redundancy positions. So \( w_1 = (0, 1, 0, 1, 1, 0, 0) \) and \( d(w_1, y) = 3 > t \). So we use the permutation \( \sigma^2 \) to get the new information set \( \{2, 3, 4, 5\} \). Applying \( \sigma^2 \) to the old parity check equations we get

\[
\begin{align*}
\alpha_6 &= \alpha_2 + \alpha_3 + \alpha_4, \\
\alpha_0 &= \alpha_2 + \alpha_3 + \alpha_5, \\
\alpha_1 &= \alpha_3 + \alpha_4 + \alpha_5.
\end{align*}
\]

So \( w_2 = (0, 0, 0, 1, 0, 1, 1) \) and \( d(w_2, y) = 1 \leq t \). So we decode \( y \) to \( w_2 \). Note \( w_2 = x = e_1 \), so \( y \) was decoded properly.

One way coding theory has been applied recently is in the “hat color problem.”

The idea of the problem is that \( n \) players sit in a circle and each is given either a red or a blue hat. Each player can see all the other players’ hats, but not her own. No communication is allowed. The object of the game is for at least one player to guess her hat color correctly, and none guess incorrectly (any player or players can refrain from guessing if he wishes). This may seem like complete chance, but R. Berlekamp has given a solution where the game can be won \( n/n+1 \) of the time. The idea of the solution for small numbers is not that difficult. Let \( n = 3 \). There is a 75% probability that there will be 2 hats of one color, and 1 of the other. If a player sees the other 2 have different color hats, she passes. If she sees that the other 2 players have the same color hat, though, she guesses that her hat is the opposite color. This plan will work 75% \((n/n+1)\) of the time.
The solution for bigger values of n requires coding theory. Berlekamp used BCH codes and other properties of coding theory. I plan to study this problem and its solution more next semester.

Error-correcting code theory is essential to our modern life. The rapid growth of the amount of information needed to be transmitted make it very important to continue our study of this subject. Codes that are more efficient to transmit, correct more errors, and are more efficient to decode are always needed. For example, a more efficient way to decode QR codes would be very helpful. With increase demands for information transfer, in addition to new uses for the subject in other areas, the importance of research in error-correcting code theory will only increase as time goes on.