Rose-Hulman Undergraduate Mathematics Journal

Volume 5
Issue 2

Article 6

On codes generated from quadratic surfaces in PG(3,q)

Mandy Passmore
University of Mary Washington, apass8rk@umw.edu

Jenny Stovall
University of Mary Washington, jstov7og@umw.edu

On codes generated from quadratic surfaces in $PG(3,q)$

Mandy Passmore and Jenny Stovall

September 23, 2004

Abstract

We construct two families of low-density parity-check codes using point-line incidence structures in $PG(3,q)$. The selection of lines for each structure relies on the geometry of the two classical quadratic surfaces in $PG(3,q)$, the hyperbolic quadric and the elliptic quadric.

Keywords: LDPC code, elliptic quadric, hyperbolic quadric
Classification: 94A10, (05B25, 51E20)

1 Introduction

With an ever-growing number of applications including deep space communication, cellular technology, and fiber optics, just to name a few, the study of error-correcting codes is becoming increasingly important. Error-correcting codes offer a way of increasing the reliability of a digital signal. The basic concept behind error-correcting codes is to use mathematical techniques to encode a "message" that the user wishes to transmit. The act of encoding a message creates a longer signal, a codeword, with the extra pieces of information used to help identify and correct any errors that may occur during the transmission of the codeword. The codeword is sent over a channel and, should errors occur (for instance by a scratch on a CD or atmospheric interference with a cellular transmission), the received message can hopefully be decoded to retrieve the original message.

The theory of error-correcting codes was originally introduced by C. E. Shannon [12] in the late 1940s. In his paper, Shannon showed (roughly) that up to a special constant called the "channel capacity," it is possible to transmit information with an arbitrarily small probability of error by using long enough codewords. Shannon used probabilistic techniques in his paper and did not provide any method for actually constructing such codes. The world was enlightened in the early '90s when a group of researchers in France produced the first codes [1] that came close to the limit set by Shannon.

Linear block codes are defined simply as a subspace of a vector space. Binary linear block codes are subspaces of a vector space over the finite field $GF(2)$. These codes are the most widely used and can be represented in several ways. The most useful representation for our purposes is through a parity-check matrix. A parity-check matrix is a matrix whose rows generate the subspace that is orthogonal to the subspace that represents our code.

In the 1960s, the concept of low-density parity-check (LDPC) codes was introduced by Gallager [3]. Quite simply, these codes are generated by a sparse parity-check matrix. The codes were forgotten for many years, but rediscovered when it was shown in [7] that these codes perform quite well under a technical decoding algorithm known as belief propagation [10]. In 2001, Fossorier, et al [6] examined a class of codes generated by incidence structures in finite geometries and showed that these codes provide wonderful examples of LDPC codes with good performance under iterative decoding. Other mathematicians have since produced other LDPC codes based on various incidence structures in discrete mathematics (see [5], [8], [11], [13], for instance). A common technique in describing these codes relies on their graphic representation, called the Tanner graph, introduced in [14].

In this paper, we examine two new classes of LDPC binary linear error-correcting codes and investigate their effectiveness in comparison to an uncoded signal holding the same information. We study the mathematical properties of these codes and provide simulation data to exhibit their effectiveness.

2 Finite Projective 3-space

We start by providing a bit of background on finite geometry. Projective geometry is formally defined as a collection of objects called points and other objects called lines with some incidence relation between them. We use the notation $PG(3,q)$ to represent the classical three-dimensional finite projective geometry of order q. There are axioms that define the space; however, due to a theorem of Veblen and Young [15], we can model finite projective 3-space using a four-dimensional vector space over $GF(q)$ with vectors of the form (w,x,y,z) for $w,x,y,z \in GF(q)$ (where q is a prime power). In this model the one-dimensional subspaces represent points, the two-dimensional subspaces represent lines, and the three-dimensional subspaces represent planes. Since scalar multiples of a non-zero vector generate the same one-dimensional space, we left-normalize our vectors to represent the points of $PG(3,q)$. This means that the first non-zero coordinate of every vector is a one, regardless of the finite field with which we are working.

In finite projective 3-space, we can use counting arguments to determine the number of points and the number of lines. For $(w,x,y,z) \in V$ we have q choices for each coordinate w, x, y, and z. However, we must subtract one from this number in order to disregard the zero vector since the zero vector does not generate a one-dimensional subspace. Finally we divide by $(q-1)$, the number of scalar multiples of any non-zero vector, yielding the final count of $\frac{q^4-1}{q-1} = q^3+q^2+q+1$ points. By counting the number of two-dimensional vector subspaces through a given one-dimensional subspace we can easily find that the number of points on a line is $q+1$ and we can use this fact to count the number of lines. Since two distinct points determine a unique line, $\binom{q^3+q^2+q+1}{2}$ is a good starting point for counting the lines; however, since every line has at least three points on it we will have counted each line multiple times. Hence, we divide this number by $\binom{q+1}{2}$, the number of ways to choose two of the $q+1$ points on any given line. Therefore, we have $\binom{q^3+q^2+q+1}{2}/\binom{q+1}{2} = (q^2+q+1)(q^2+1)$ lines. We can use similar arguments to show that the number of planes in finite projective 3-space is q^3+q^2+q+1, the same as the number of points.

An important characteristic of projective geometry (in comparison to Euclidean geometry) is that there are no parallel lines. Hence, if two lines lie in a common plane, they must intersect in a point. Notice that our model follows this axiom. If two distinct two-dimensional subspaces (i.e., two distinct lines) lie in a common three-dimensional subspace (a plane), then they must meet in a one-dimensional subspace (a point). This is the distinguishing feature of projective geometry and one which we will make use of down the road. Also, since the coordinates of our geometry are based on finite fields, there is no notion of distance in a finite projective space. Hence, we never make use of the notion of angle, distance, length, etc. when working in finite projective spaces. Our geometry will be based solely on three objects, points, lines, and planes, and incidences between them.

3 Quadratic Surfaces of $PG(3, q)$

Just as we do in Euclidean space, we can consider sets of points that satisfy a quadratic equation. For instance, in the real plane, we have four "nondegenerate quadratics," the conic sections. (Can you name them?) In $PG(3, q)$, there are just two nondegenerate quadratics. In projective spaces, we must modify the definition of quadratic slightly (to that of a *quadratic form*). We will not address this in detail, but rather jump right to the heart of the matter.

There are two quadratic surfaces that live in $PG(3,q)$, the so-called hyperbolic quadric, and the elliptic quadric. We form the hyperbolic quadric, $\mathcal{H}$, by looking at normalized vectors (w,x,y,z) representing points of $PG(3,q)$ that satisfy the condition $wy = xz$. We can count the number of points that satisfy this condition. If $w = x = y = 0$, then z must be one so there is only one vector, $(0,0,0,1)$, of this form. If $w = x = 0$ and $y \neq 0$, then y must be one and there are q choices for z. Also, if $w = 0$ and $x \neq 0$ then x must be one, z must be zero, and there are q choices for y. Finally, when $w \neq 0$, $w = 1$, there will be q choices for any two of the three remaining coordinates, x, y, or z, and the third will be uniquely determined by those two, yielding q^2 points. Combining all of these cases, we have a total of $q^2 + 2q + 1 = (q+1)^2$ points in $\mathcal{H}$. Lines that lie in $\mathcal{H}$ are two-dimensional subspaces of the form $l_b = \{\langle(1,0,0,b),(0,1,b,0)\rangle : b \in GF(q)\} \cup \{\langle(0,0,0,1),(0,0,1,0)\rangle\}$ or of the form $m_b = \{\langle(1,b,0,0),(0,0,b,1)\rangle : b \in GF(q)\} \cup \{\langle(0,1,0,0),(0,0,1,0)\rangle\}$. One can easily check that any two lines of the form l_i are skew. Also any two lines of the form m_j are skew. Finally, any line of the form l_i meets every line of the form m_j in a unique point. For a line l_i, we know that there are $q+1$ points that lie on it. Also, for each of the $q+1$ points on l_i, there is a line m_j running through it, also containing $q+1$ points. Thus we observe that the $(q+1)^2$ total points in the hyperbolic quadric are ruled by these two families of lines (i.e., every point lies on exactly one line from each of these families). Since there are $q+1$ lines of the form l_i and $q+1$ lines of the form m_j, the total number of lines is $2q+2$. It is not hard to check that no other lines of $PG(3,q)$ lie in $\mathcal{H}$. While we can discuss numerous properties of the hyperbolic quadric, the complex geometry of the elliptic quadric strictly limits our discussion of its characteristics.

We form the elliptic quadric $\mathcal{E}$ by looking at vectors (w,x,y,z) that satisfy the condition $dw^2 + wx + x^2 + yz = 0$ where $1 - 4d$ is a non-square when q is odd. When q is even, a trace condition on d gives us the non-degenerate elliptic quadric. The trace condition is not pertinent to this discussion but the interested reader can find further detail in [4]. One can show that $\mathcal{E}$ has

q^2+1 points and exactly one tangent plane at any point on $\mathcal{E}$. Moreover, no three points of $\mathcal{E}$ are collinear. The cross section resulting when $\mathcal{E}$ is cut by a plane is an oval, a set of $q+1$ planar points also with the property that no three points are collinear.

Using the properties of these two quadratic surfaces we generate low-density parity-check codes as discussed below, beginning with the hyperbolic quadric.

4 Codes generated by a hyperbolic quadric

In order to generate codes from the quadratic surfaces, we first introduce the notion of an *incidence matrix*. We construct an incidence matrix for $\mathcal{H}$ by labeling the columns of a matrix with the $(q+1)^2$ points of $\mathcal{H}$ and the rows with the $2q+2$ lines. We place a one in a position ij if the line corresponding to row i runs through the point corresponding to column j, and a zero in that position otherwise. We note that any arithmetic involving the matrix is performed modulo 2. We then obtain a matrix with row weight $q+1$, since there are $q+1$ points on every line, and column weight two, since two lines (one of the form l_i and one of the form m_j) run through every point. This yields a sparse matrix which we will use to generate a low-density parity-check code. We use $\mathcal{C}_{\mathcal{H}_q}$ to denote this LDPC code generated by the hyperbolic quadric of $PG(3,q)$. Similarly, we use $H_{\mathcal{H}_q}$ to denote its corresponding parity-check matrix.

Example 4.1 *Let $q=2$. Then the hyperbolic quadric contains nine points*

$$(0,0,0,1),(0,0,1,0),(0,1,0,0),(1,0,0,0),(0,0,1,1),$$

$$(1,1,0,0),(1,0,0,1),(0,1,1,0),(1,1,1,1).$$

There are three points on a line and a total of six lines. Using these nine points and six lines we create the following incidence matrix which will be *our parity-check matrix:*

$$H_{\mathcal{H}_2} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

Each row has weight three (the number of points on a line) and each column has weight two (the number of lines through a point).

For every low-density parity-check code, there is a corresponding bipartite graph called the Tanner graph. Our hyperbolic quadric creates a bipartite graph with one partition class representing points, the other representing lines, and edges determined by incidence. The girth of this graph is the length of the shortest cycle. There is evidence that it is ideal for this girth to be high (at least six) for a code to be efficient under iterative decoding. The hyperbolic quadric is of special interest because we can show that the resulting bipartite graph does not contain any cycles of length six or less, and so has girth at least eight.

Lemma 4.2 *The lines of a hyperbolic quadric never form a triangle.*

Proof: Suppose there exists a triangle T in our hyperbolic quadric. Then T contains three lines and since we know that our lines are only of two forms, l_i and m_j, by the pigeonhole principle two of these three lines must be of the same form. However, this is a contradiction since we know that no two lines of the form l_i can meet and likewise for lines m_j. Thus we have shown that the hyperbolic quadric contains no triangles. ■

5 Mathematical properties of $\mathcal{C}_{\mathcal{H}_q}$

When q is relatively small, the software package *Magma* [2] was used to find the coding parameters, the mathematical properties that are used to define the code, for the codes generated by the hyperbolic quadric. These parameters are the length, dimension, and minimum distance of the code and are listed in that order to identify a code. The length, commonly denoted as n, is exactly as it sounds, the number of bits, both information bits and extra decoding bits, in each codeword from that code. The dimension, k, of the code indicates the number of bits within each codeword that contain information, as opposed to the additional bits used in decoding. Thus, if H is the parity-check matrix of the code then $k = n - rank(H)$. Finally, the minimum distance, d, of a code is the minimum number of bits that differ when comparing any two codewords. A codeword with parameters n, k, d is usually denoted by $[n, k, d]$. A summary of the data found for these codes is given in Table 1. We noticed an obvious pattern in the dimension and minimum distance of the codes in the table.

The geometry of the hyperbolic quadric is quite structured. As a result, the geometry lends a hand in proving properties of the associated code. We were able to prove that the code formed from the hyperbolic quadric is a $[(q+1)^2, q^2, 4]$-code.

Lemma 5.1 *$\mathcal{C}_{\mathcal{H}_q}$ has dimension q^2.*

Proof: Consider H, the parity-check matrix. To find the dimension of our codes we will first try to create the smallest possible linearly dependent set of rows of H. Let B be a basis for the row space of H (consisting of rows of H) and suppose the row corresponding to l_1 is in B. Then we know that there will be $q + 1$ ones in the row corresponding to l_1. In order to cancel these ones out to obtain the zero vector, we need to include all $q + 1$ of the rows corresponding to the lines m_j (which run through the $q + 1$ points on l_1). However, in including these lines we have covered all $(q + 1)^2$ points in the hyperbolic quadric and there are $(q + 1)^2 - (q + 1) = q(q + 1)$ ones in

q	length	dimension	minimum distance
2	9	4	4
3	16	9	4
4	25	16	4
5	36	25	4
7	64	49	4
8	81	64	4
9	100	81	4
13	196	169	4
16	289	256	4
17	324	289	4
19	400	361	4
23	576	529	4

Table 1: Parameters for $\mathcal{C}_{\mathcal{H}_q}$

the vectors of B that have not been canceled out to yield the zero vector. Thus we now must include the remaining lines of the form l_i, $i \neq 1$ each containing $q+1$ points to give us the $q(q+1)$ ones needed to cancel those mentioned above. Only in this manner will we obtain the zero vector giving us a linearly dependent set. However, this set of vectors contains every row of our parity-check matrix. Thus the largest linearly independent set is found by taking only one of these vectors out. But l_1 was completely arbitrary. Thus any $2q+1$ rows of H are linearly independent. Now, recall the dimension theorem from linear algebra, which states $dim(V) = dim(S) + dim(S^{\perp})$ for any subspace S of V. Let S be our code $\mathcal{C}$. The dimension of V is $(q+1)^2$ and we found that the dimension of $S^{\perp}$ is $2q+1$. Thus we have $q^2 + 2q + 1 = dim(S) + 2q + 1$, or equivalently $q^2 = dim(S)$. ■

Lemma 5.2 *$\mathcal{C}_{\mathcal{H}_q}$ has minimum distance four.*

Proof: Let H_i represent the i^{th} column of H, the parity-check matrix, for $i = 1, 2, \ldots, (q+1)^2$. Let $c = (c_1, c_2, \ldots, c_{(q+1)^2})$ be a codeword. Then $Hc^{\perp} = 0$, or equivalently $\sum_i c_i H_i = 0$. We will now consider the existence of codewords of small weight. Suppose c is a codeword of weight one and suppose the one occurs in the i^{th} position of c. Then, when c is multiplied by the parity-check matrix, the result will be the corresponding column H_i, which we know has weight two, and thus cannot be equal to the zero vector. Hence there is no codeword of weight one.

Suppose c is a codeword of weight two. Then we know that c_i and c_j, for some $i \neq j$, contain a one, and all other positions contain zeros. When we multiply this codeword c by the parity-check matrix we obtain zero. This implies that $H_i + H_j = 0$. However, in order for $H_i + H_j$ to equal zero, H_i and H_j must have ones in the same positions, and so be equal. But we assumed $i \neq j$ and that each column represents a distinct point. Therefore, it is impossible to have a codeword of weight two.

Now suppose c is a codeword of weight three. Then by similar arguments as above there exist three columns $H_i, H_j, H_k, i \neq j, j \neq k, k \neq i$ such that $H_i + H_j + H_k = 0$. In order for this sum to be 0, each pair of lines must go through a common point so that two ones can sum to zero. The only way for each pair of lines to go through a common point is if the lines are concurrent or the lines form a triangle. Clearly the lines cannot be concurrent since any point of $\mathcal{H}$ has exactly two lines through it. As we discussed in Lemma 4.2, the hyperbolic quadric contains no triangles. Hence it is impossible to have a codeword of weight three. We have shown that the minimum weight of our code cannot be two or three. However, we can construct a codeword of weight four as follows.

We begin by choosing a quadrangle of $\mathcal{H}$ and letting P_i, P_j, P_k, P_l be the points of this quadrangle. These points are chosen so that P_i, P_j are on a line of the form l_i as are P_k and P_l. Similarly, P_i, P_k are on a line of the form m_j as are P_j and P_l. Then let c be the characteristic vector corresponding to the points on this quadrangle. In other words, c is the vector with ones in positions i, j, k, and l and zeros in all other positions. Since every line of $\mathcal{H}$ contains either zero or two points of the quadrangle, the sum of these columns is zero. Hence, we have found a codeword of weight four. We can extend this argument to find numerous other examples of weight four codewords depending on which quadrangle is chosen. Thus the minimum distance of our code is four. ■

Theorem 5.3 *$\mathcal{C}_{\mathcal{H}_q}$ is a $[(q+1)^2, q^2, 4]$ code.*

Proof: The proof of this theorem follows from Lemmas 5.1 and 5.2. ■

6 Codes generated by an elliptic quadric

We now examine a class of codes generated by an elliptic quadric. For our discussion, we will be using several properties of the elliptic quadric that can be found in [4]. We chose to consider points and lines not incident with $\mathcal{E}$. We can easily count the number of points off $\mathcal{E}$. There are $(q+1)(q^2+1)$ points in 3-space, q^2+1 of which are on the elliptic quadric. Hence, there are $q(q^2+1)$ points off $\mathcal{E}$. We can again use a simple counting argument to find the number of lines that are skew to $\mathcal{E}$. We know that there are q^2+1 planes tangent to $\mathcal{E}$ (one for each point) and there are $q+1$ lines through a point in each such plane, yielding a total of $(q+1)(q^2+1)$ lines that meet $\mathcal{E}$ in exactly one point. Also, since any two points determine a line in $PG(3,q)$, there are $\binom{q^2+1}{2}$ lines that meet $\mathcal{E}$ in two points. No lines can meet $\mathcal{E}$ in three points. Now recall that the total number of lines in $PG(3,q)$ is $(q^2+q+1)(q^2+1)$. Thus we find that the number of lines skew to $\mathcal{E}$ is $(q^2+q+1)(q^2+1) - \binom{q^2+1}{2} - (q+1)(q^2+1) = \frac{1}{2}q^2(q^2+1)$.

We form codes $\mathcal{C}_{\mathcal{E}_q}$ by using the points off $\mathcal{E}$ along with all lines skew to $\mathcal{E}$ to create an incidence matrix that will be our parity-check matrix $H_{\mathcal{E}_q}$. We begin by labeling the columns of $H_{\mathcal{E}_q}$ with the $\frac{1}{2}q^2(q^2+1)$ lines and the rows with the $q(q^2+1)$ points (note that we swapped the roles of points and

q	length	dimension
3	45	17
5	325	196
7	1225	877
9	3321	2584
11	7381	6041
13	14365	12156
17	41905	36976

Table 2: Parameters for $\mathcal{C}_{\mathcal{E}_q}$, q odd

lines in our matrix). Hence, the column weight of $H_{\mathcal{E}_q}$ is $q+1$ since there are $q+1$ points on a line in $PG(3,q)$. We can also find the row weight, the number of lines through a point, by multiplying the number of lines by the number of points on a line and dividing by the number of points off of $\mathcal{E}$ as follows: $\frac{1}{2}q^2(q^2+1)(q+1)/(q^3+q) = \frac{1}{2}q(q+1)$. Note that this follows because the number of lines through a point is a constant (this is a property of the elliptic quadric which can be found in [4]). Note that the row weight will be odd if $q \equiv 1 \pmod 4$ and even if $q \equiv 3 \pmod 4$. A summary of the coding parameters obtained through *Magma* computations, when q is odd, is given in Table 2. We observe a pattern based on this data and can form a conjecture about the dimension k of the code $C_{\mathcal{E}_q}$ when q is odd.

Conjecture 6.1 *For q odd, $\mathcal{C}_{\mathcal{E}_q}$ has dimension $k = n - (q^3+q) + \delta$ where $\delta = 1$ if $q \equiv 1 \pmod 4$ and $\delta = 2$ if $q \equiv 3 \pmod 4$.*

Using this conjecture, we can actually prove a bound on the dimension of these codes. We observe that when q is odd, the sum of the rows of $H_{\mathcal{E}_q}$ is the zero vector. This follows from the fact that the column weight, $q+1$, is even in this case and leads us to the following proposition (which agrees with the conjecture).

Proposition 6.2 *For q odd, the dimension of $\mathcal{C}_{\mathcal{E}_q}$ is at least $n-(q^3+q)+1$.*

Proof: The rows of $H_{\mathcal{E}_q}$ in this case sum to the zero vector. Hence, the rank of $H_{\mathcal{E}_q}$ is at most q^3+q-1. Therefore, by the dimension theorem, the dimension of $\mathcal{C}_{\mathcal{E}_q}$ is at least $n-(q^3+q)+1$. ■

Example 6.3 *Let $q=3$. Then the number of points off $\mathcal{E}$ is $q(q^2+1)=30$ and the number of lines skew to $\mathcal{E}$ is $\frac{1}{2}q^2(q^2+1)=45$. Thus the parity-check matrix is a 30×45 matrix. We know that the column weight is $q+1=4$ and the row weight is $\frac{1}{2}q(q+1)=6$.* Magma *confirmed that, in accordance with our conjecture the dimension, k, is 17.*

A summary of the data found for codes when q is even is given in Table 3. Note that we have no conjecture on the dimension for these codes when q is even.

Example 6.4 *When $q=2$, the number of points off $\mathcal{E}$ is $q(q^2+1)=10$ and the number of lines skew to $\mathcal{E}$ is $\frac{1}{2}q^2(q^2+1)=10$. Hence the parity-check matrix is a 10×10 matrix.* Magma *was used to find this matrix explicitly, and we provide it here.*

$$H_{\mathcal{E}_2}=\begin{bmatrix}
1&1&0&0&1&0&0&0&0&0\\
0&0&0&1&1&0&0&0&1&0\\
1&0&1&0&0&0&1&0&0&0\\
0&1&0&1&0&1&0&0&0&0\\
0&0&0&1&0&0&0&1&0&1\\
0&0&0&0&0&1&1&1&0&0\\
0&0&1&0&0&0&0&1&1&0\\
1&0&0&0&0&1&0&0&1&0\\
0&1&0&0&0&0&1&0&0&1\\
0&0&1&0&1&0&0&0&0&1
\end{bmatrix}$$

The column weight of the parity-check matrix is $q+1=3$ and the row weight is $\frac{1}{2}q(q+1)=3$. Using the software package Magma *[2] we found the dimension, k, is four. Also we found the minimum distance to be four.*

q	length	dimension
2	10	4
4	136	92
8	2080	1774
16	32896	30352

Table 3: Parameters for $\mathcal{C}_{\mathcal{E}_q}$, q even

q	minimum distance
2	4
3	8
4	6

Table 4: Minimum distances for $\mathcal{C}_{\mathcal{E}_q}$

When q is really small, the software package *Magma* [2] was used to find the minimum distance for codes from the elliptic quadric. We were only able to find data for very small values of q due to the computation time required for larger values. A summary of the data found for these small values of q is given in Table 4.

7 Simulating the codes

In order to test the performance of our codes, we ran simulations using R.H. Morelos-Zaragoza's iterative probabilistic decoding of linear block codes [9] on the 72 node parallel-processing and visualization system now known as the"Immersive Visualization System"(IVS) operated by the Department of Mathematics and Statistics at James Madison University[1]. We ran 500,000

[1] a special thanks to Jim Sochacki and Josh Blake for their help in running these simulations and making their system available to us

or more codewords through the simulation code depending on the length of the code. We wanted to test codes of length about 50, 100, and 500 from both the hyperbolic and elliptic quadrics. We ran the simulations with a signal-to-noise ratio ranging from one to six and compared the results to those of an uncoded message to observe how the codes performed, that is, how accurately the code detected and fixed any errors that occurred. As the signal-to-noise ratio increases, that is, the signal grows stronger than the noise that might interfere with it, the performance of the coded signal greatly improves. On the graphs, the x-axis represents the signal-to-noise ratio while the y-axis represents the number of bit errors per codeword that go uncorrected. Hence, the better the performance of the code, the lower the graph of the coded message will drop.

The results of the simulations are given in the Excel charts below. We note that the simulation results for the codes $\mathcal{C}_{\mathcal{E}_q}$ are quite good. This could be due to the complex geometric nature of the incidence structure and the extreme sparseness of the parity-check matrix. The codes $\mathcal{C}_{\mathcal{H}_q}$, on the other hand, do not perform as well. The column weight of the parity-check matrix for the codes of the elliptic quadric grows with q, whereas the column weight of the hyperbolic quadric is a constant. This could explain the disparity in the performance of the two families of codes.

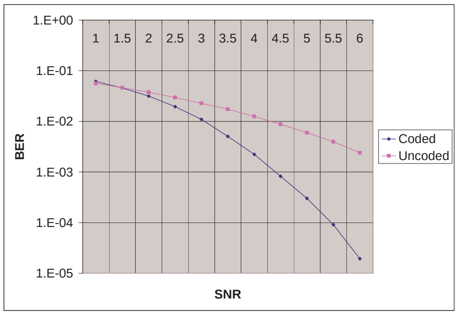

Figure 1: Performance of $[64, 49, 4]$- code $\mathcal{C}_{\mathcal{H}_7}$

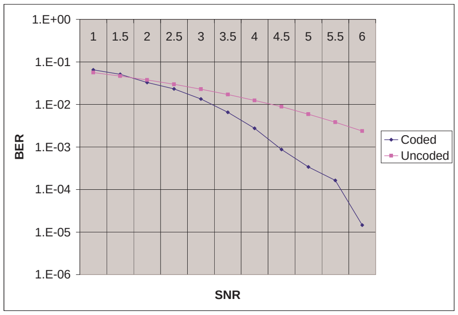

Figure 2: Performance of $[100, 81, 4]$- code $\mathcal{C}_{\mathcal{H}_9}$

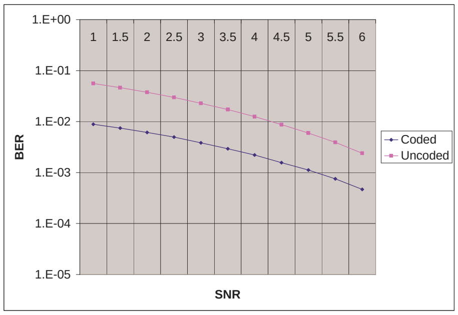

Figure 3: Performance of $[576, 529, 4]$- code $\mathcal{C}_{\mathcal{H}_{23}}$

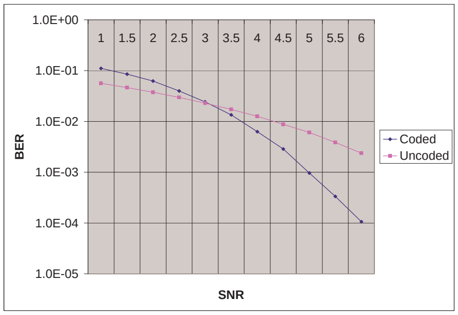

Figure 4: Performance of $[45, 17, 8]$- code $\mathcal{C}_{\mathcal{E}_3}$

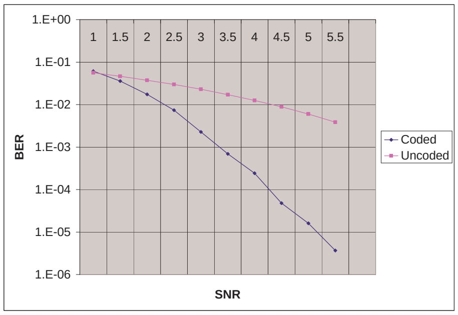

Figure 5: Performance of $[136, 92, 6]$- code $\mathcal{C}_{\mathcal{E}_4}$

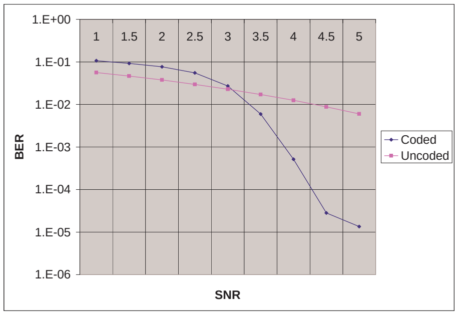

Figure 6: Performance of $[325, 196]$- code $\mathcal{C}_{\mathcal{E}_5}$

8 Conclusion

Through our research we found that the use of quadratic surfaces in 3-space proves to be an effective method of code construction. The geometry-based structure of these codes makes them appealing and provides a means for proving mathematical properties about the codes. While the hyperbolic quadric is a simpler construction the elliptic quadric outperforms it in practice. Even the codes $\mathcal{C}_{\mathcal{E}_q}$ of shorter lengths perform competitively. Although we have no simulation results for codes of practical lengths (1000 or more), we can hope that longer length codes will perform well in the real world.

Acknowledgment: This research was conducted as part of the Summer Science Institute under the advisement of Dr. Keith E. Mellinger at the University of Mary Washington during the summer of 2004. We would like to thank the program for their support.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo codes," *Proc. of the 1993 IEEE Internat. Communications Conf.*, Geneva, Switzerland (May 23–26, 1993), 1064–1070.

[2] J. Cannon and C. Playoust, "An Introduction to Magma", University of Sydney, Sydney, Australia (1994).

[3] R. G. Gallager, Low density parity check codes, *IRE Trans. Infom.Theory*, **IT-8**, (1962), 21–28.

[4] J.W.P. Hirschfeld, "Projective Geometries over Finite Fields," Oxford University Press, second edition (1998).

[5] J.-L. Kim, U. N. Peled, I. Perepelitsa and V. Pless, Explicit construction of families of LDPC codes of girth at least six, *Proc. 40th Allerton*

Conf. on Communication, Control and Computing, P. G. Voulgaris and R. Srikant, Eds. (Oct. 2–4, 2002) 1024–1031.

[6] Y. Kuo, S. Lin and M. P. C. Fossorier, Low-density parity-check codes based on finite geometries: a rediscovery and new results, *IEEE Trans. Inform. Theory*, **47** no. 7 (2001), 2711–2736.

[7] D. J. C. MacKay and R. M. Neal, Near Shannon limit performance of low density parity check codes, *Electron. Lett.*, **32** no. 18 (1996), 1645–1646.

[8] K. E. Mellinger, LDPC Codes and triangle-free line sets, *Designs, Codes, and Cryptog.*, **32**: 1-3 (2004) 341 - 350.

[9] R.H. Morelos-Zaragoza, The Art of Error Correcting Coding, Wiley, 2002.

[10] J. Pearl, *Probabilistic Reasoning in Intelligent Systems,* 2nd ed. San Francisco, CA: Kaufmann, 1988.

[11] J. Rosenthal and P. O. Vontobel, Constructions of LDPC codes using Ramanujan graphs and ideas from Margulis, in *Proc. of the 38th Annual Allerton Conference on Communication, Control, and Computing*, (2000) 248-257.

[12] C. E. Shannon, A mathematical theory of communication, *Bell Syst. Tech. J.* **27** (1948), 379–423, 623–656.

[13] M. Sipser and D. A. Spielman, Expander codes, *IEEE Trans. Inform. Theory* **42** no. 6, (1996) 1710–1722.

[14] R. M. Tanner, A recursive approach to low complexity codes, *IEEE Trans. Inform. Theory* **IT - 27** (1981), 533–547.

[15] O. Veblen and J. W. Young. *Projective Geometry*, vol I. Ginn, Boston, 1910.