MSRI-UP 2009 PROJECT TOPIC IDEAS

JOHN LITTLE – COLLEGE OF THE HOLY CROSS

1. Toric Codes

A first group of project topics deals with a class of codes known as *toric codes*.

General Background. In [6], [7], J. Hansen introduced these codes using some constructions from algebraic geometry (toric varieties). In somewhat more down-to-earth language than he used, let $P \subset \mathbb{R}^m$ be an integral convex polytope (the convex hull of some set of integer lattice points). (The book [18] as a good general reference for the geometry of polytopes.) Suppose that $P \cap \mathbb{Z}^m$ is properly contained in the rectangular box $[0, q-2]^m$ (which we denote $\square_{q-1}$), for some prime power q. Then a toric code is obtained by evaluating linear combinations of the monomials with exponent vector in $P \cap \mathbb{Z}^m$ at some subset (usually all) of the points of $(\mathbb{F}_q^*)^m$. We formalize this in the following definition.

Definition 1.1. *Let $\mathbb{F}_q$ be a finite field with primitive element α. For $f \in \mathbb{Z}^m$ with $0 \leq f_i \leq q-2$ for all i, let $p_f = (\alpha^{f_1}, \ldots, \alpha^{f_m})$ in $(\mathbb{F}_q^*)^m$. For any $e = (e_1, \ldots, e_m) \in P \cap \mathbb{Z}^m$, let x^e be the corresponding monomial and write*

$$(p_f)^e = (\alpha^{f_1})^{e_1} \cdots (\alpha^{f_m})^{e_m}.$$

The toric code $C_P(\mathbb{F}_q)$ over the field $\mathbb{F}_q$ associated to P is the linear code of block length $n = (q-1)^m$ with generator matrix

$$G = ((p_f)^e),$$

where the rows are indexed by the $e \in P \cap \mathbb{Z}^m$, and the columns are indexed by the $p_f \in (\mathbb{F}_q^*)^m$. In other words, letting $L = \mathrm{Span}\{x^e : e \in P \cap \mathbb{Z}^m\}$, we define the evaluation mapping

$$\begin{aligned} \mathrm{ev} : L &\rightarrow \mathbb{F}_q^{(q-1)^m} \\ g &\mapsto (g(p_f) : g \in (\mathbb{F}_q^*)^m) \end{aligned}$$

Then $C_P(\mathbb{F}_q) = \mathrm{ev}(L)$. If the field is clear from the context, we will often omit it in the notation and simply write C_P. The matrix G will be called the standard generator matrix for the toric code.

If P is the interval $[0, \ell-1] \subset \mathbb{R}$, then $C_P(\mathbb{F}_q)$ is the Reed-Solomon code $RS(\ell, q)$ that we studied in the first part of the seminar. So toric codes are, in a sense, generalizations of Reed-Solomon codes. Not all toric codes are good from the coding theory perspective, but the class does contain some very good codes. The connections with convex geometry (theory of polytopes) and algebraic geometry are another reason for the interest in this topic.

Date: April 30, 2009.

Some of the properties of general toric codes were studied in [4] from the 2001 SIMU program at the University of Puerto Rico at Humacao. Several of this group's results were later expanded by D. Ruano in [16].

One important operation on polytopes is the so-called *Minkowski sum*. The Minkowski sum of two polytopes $P_1 + P_2$ is just the set of vector sums of points in the two polytopes:

$$P_1 + P_2 = \{p_1 + p_2 \mid p_i \in P_i\}.$$

For instance, if P_i is the line segment from 0 to e_i in the plane, then $P = P_1 + P_2$ is the unit square $[0, 1] \times [0, 1]$. A restricted but interesting class of polytopes are the *zonotopes*. A zonotope is a polytope that is the Minkowski sum of a finite number of primitive lattice segments (the *generators*). For instance the unit square, unit cube, etc. are zonotopes. But there are lots of even more interesting ones too!

Project 1 – Systematic "census" of (generalized) toric codes. One of the slightly surprising thing about toric codes is that there are so many different examples (with drastically differing properties). Any lattice polytope $P \subset \square_{q-1}$ is "fair game." Still, there are some general ways one can think of reducing the complexity of classifying the possibilities. For instance, in [13] and [16] it is shown that if two polytopes P, Q are *lattice equivalent* (that is, there is some invertible integer matrix A whose inverse is also an integer matrix and some integer vector v such that the affine mapping $T(x) = Ax + v$ satisfies $T(P) = Q$), then the codes $C_P(\mathbb{F}_q)$ and $C_Q(\mathbb{F}_q)$ are *monomially equivalent*. This notion of equivalence for codes means that the parameters (including the minimum distances) must be the same.

This is a very nice statement, but it is still somewhat unsatisfactory since "most" affine mappings T as in the previous paragraph will take polytopes $P \subset \square_{q-1}$ to $Q = T(P)$ with $Q \not\subset \square_{q-1}$.

The lattice points in a polytope $P \subset \square_{q-1}$ are used as exponents in monomials which are then evaluated at elements of $(\mathbb{F}_q^*)^m$ to get the entries in the codewords of our toric code. Since $x^{q-1} = 1$ for all $x \in \mathbb{F}_q^*$ (Lagrange's theorem for finite groups), it also makes sense to consider the exponents as vectors of integers mod $q-1$. Then it can also be shown (you should start by doing this!) that if $B \in \mathrm{GL}(m, \mathbb{Z}_{q-1})$– that is, B is a matrix with entries in the integers mod $q-1$ which has a matrix inverse in the ring of matrices with entries in $\mathbb{Z}_{q-1}$–then the toric codes formed from $P \cap \mathbb{Z}_{q-1}^m$ and from $B(P \cap \mathbb{Z}_{q-1}^m)$ are again *monomially equivalent*.

Now this is again slightly unsatisfactory because it is easy to find examples of P and B where the set $P \cap \mathbb{Z}_{q-1}^m$ is mapped to a set of points in $\mathbb{Z}_{q-1}^m$ that is not equal to $Q \cap \mathbb{Z}_{q-1}^m$ for any convex polytope Q. So we need to extend our point of view to study what the article [16] calls *generalized* toric codes – the definition is the same except that the monomials that are evaluated come from *any subset* of $\square_{q-1}$, not just from the lattice points in a convex polytope.

The big advantage of this approach, though is that it lets you apply the well-developed machinery of *group actions on finite sets* to study and count the orbits under the action of $\mathrm{GL}(m, \mathbb{Z}_{q-1})$. The main foci of this project would be to

(1) Learn (or review) the basics of enumeration of orbits under finite group actions on finite sets (the cycle index polynomial, etc.) – see [1] for a presentation geared toward applications in coding theory.
(2) Apply this to the enumeration of equivalence classes of (generalized) toric codes in some relatively manageable cases, say over the field $\mathbb{F}_q$ with $q =$

$4, 5, 7, 8, 9, 16$ in the cases $m = 2$ or maybe $m = 3$, and k either relatively small or relatively close to $(q-1)^m$. There is a symmetry involved here coming from the pairing of a code C and its dual code $C^\perp$. Moreover, the structure of C completely determines the structure of $C^\perp$ and vice versa. For the weight distribution of the codewords, for instance, this comes from the *MacWilliams identities* – see Chapter 7 of [8].

(3) Generate and study the corresponding (generalized) toric codes. There are some quite good codes to be found in places here – in some cases as good or better than any codes known as of this writing (see the online tables of Markus Grassl, [5]). See the example in the description of Project 3 below, for instance.

Comments and Notes:

(1) There is probably enough to do here for *two or more groups* to work on different parts of this.
(2) I expect this project would be pretty heavily computational after the original phase. *Magma* is the software of choice for most of the operations needed here (both for working with the group actions and for the coding theory computations of computing minimum distances, etc.) It is possible, but tedious, to do small computations via the online "Magma calculator." But that is limited to 20 seconds CPU time per computation. Need to check whether MSRI has a license for Magma(!) It would also help to have a really fast/powerful computer to use for some things.
(3) David Joyner has written a collection of Magma procedures for toric codes, see his web page [10]. These might be useful for several of the other projects as well.

Project 2 – Minkowski length of polytopes in $\mathbb{R}^3$ and minimum distance of $m = 3$ toric codes. Starting with [12] and continued and expanded by the work in [17], it has become clear that one good way to understand the minimum distance of a toric code $C_P(\mathbb{F}_q)$ (at least for q sufficiently large) is to study the decompositions of the polytope P and its subpolytopes $Q \subset P$ as *Minkowski sums*.

For instance, let P be the unit square written as $P = P_1 + P_2$, where P_1 is the line segment from $(0,0)$ to $(1,0)$ and P_2 is the line segment from $(0,0)$ to $(0,1)$. All the words in the toric code $C_P(\mathbb{F}_q)$ come from evaluating polynomials whose exponent vectors are lattice points in the square, hence of the form

$$g(x,y) = c_0 + c_1 x + c_2 y + c_3 xy.$$

The minimum weight words in the toric code $C_P(\mathbb{F}_q)$ come from evaluating polynomials that *factor* as

$$g(x,y) = (x-a)(y-b)$$

so the $x-a$ corresponds to the Minkowski summand P_1 and $y-b$ corresponds to the Minkowski summand P_2. This means that the minimum distance is

$$d(C(\mathbb{F}_q)) = (q-1)^2 - (2(q-1)-1) = (q-2)^2,$$

since reducible f as above have exactly $2(q-1)-1$ zeroes in $(\mathbb{F}_q^*)^2$.

The *full Minkowski length* of a polytope is defined in [17] as

$$\ell(P) = \max_{Q \subseteq P} \{\ell \mid Q = Q_1 + \cdots + Q_\ell \text{ for some polytopes } Q_i\}.$$

Most of the work so far using these ideas has dealt with *toric surface codes*, or equivalently the case $m = 2$ in the general definition. So the first topic here would be to try to extend some of the ideas from [12] and [17] to the case $m = 3$ (with polytopes in $\mathbb{R}^3$). In particular, some good starting points are the following questions:

(1) In any maximal Minkowski decomposition of a $Q \subset P$, the summands must be polytopes of full Minkowski length 1 (that is, polytopes that cannot be broken down any farther). *What are the polytopes of full Minkowski length equal to 1 in* $\mathbb{R}^3$*?* (Ideally, we would like a full classification up to lattice isomorphism, but nontrivial and interesting examples would be valuable too.) The ones of dimension 1 and 2 would be the same as the ones given in [17]. What are the strictly three-dimensional ones? What do the corresponding toric codes look like? Are they all good codes? Note that 3-simplices like the convex hull of $0, e_1, e_2, e_3$ are examples. Are all such polytopes simplices?
(2) Can the results of [17] be extended to give a more or less complete classification of all maximally Minkowski-decomposable polytopes in $\mathbb{R}^3$?
(3) In [17], Soprunov and Soprunova mention *zonotopes* several times. What can you say about minimum distance of toric codes from zonotopes?
(4) If you take a 3-dimensional zonotope with a relatively large number of generators in "random" directions, the resulting polytope typically gets rather round-looking (i.e. approximately spherical). Do these zonotopes give good codes? How can you measure that?
(5) Are other three-dimensional polytopes possibly better candidates for constructing good toric codes?

Project 3 – Rethinking the Minkowski length. This is probably the most speculative topic in this group. The results of [17] are really guided by the theory of lattice polytopes. As we saw in the description of Project 1, it is perhaps more natural from the point of view of toric codes to think of the lattice points in $P \subset \square_{q-1} \subset \mathbb{R}^m$ as vectors in $(\mathbb{Z}_{q-1})^m$ and drop the requirement that the set is $P \cap \mathbb{Z}^m$ for a convex polytope. In fact, most of the really good examples I know consist of *subsets* of the points in $P \cap \mathbb{Z}^m$ that have "good" properties in some sense. For instance, consider the toric code over $\mathbb{F}_8$ for the set S pictured with filled circles in Figure 1, at the top of the next page. This gives a (generalized) toric code with $m = 2$, $k = 7$, and $d = 35$ over $\mathbb{F}_8$. According to [5], this is as good as the best known code for this k and $n = 49$ over $\mathbb{F}_8$.

Now the point is that the convex hull of these points has $k = 10$ lattice points. But the minimum distance for the $k = 10$ code is *only* $d = 21$. So, somehow, leaving out the three monomials represented by the open circles gives a much better d. It is possible to see what is happening here by thinking of full Minkowski length of the convex hull (the polygon shown in Figure 1) and the points that are left when we remove the three lattice points represented by the open circles (say if you put the left hand point on the boundary at $(0, 0)$). But it would be nice to have a form of this phrased strictly in terms of subsets of $(\mathbb{Z}_7)^2$ as well.

For this project, you would try to address the following questions.

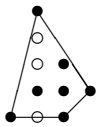

FIGURE 1. A toric $[49, 7, 35]$ code over $\mathbb{F}_8$

(1) What is the correct analog of the full Minkowski length for a subset $S \subset (\mathbb{Z}_{q-1})^2$, or $\subset (\mathbb{Z}_{q-1})^m$ in general?
(2) Can you come up with a definition that allows you to prove statements like the main theorems from [17] or [12], but without reference to a convex polytope?
(3) Does this depend on the arithmetic of $q-1$? (Note that when $q = 8$, $q - 1 = 7$ is a prime so $\mathbb{Z}_7$ is a field too. That will not always be true but it is sometimes true, as for $q = 2^5 = 32$, $q = 2^7 = 128$, and in general when $2^n - 1$ is a Mersenne prime.)
(4) Assuming a good answer to the first parts is found, can you use these ideas to *design* good toric codes?

Project 4 – Multivariate Vandermonde determinants and toric codes. A different approach to studying the minimum distance of toric codes was used in the article [13], which gives complete results for the toric codes from *rectangular solids* and *simplices* of the form $\text{conv}(0, \ell e_1, \ldots, \ell e_n)$ for integers ℓ.

The key tool in [13] is the study of determinants of maximal square submatrices of the standard generator matrix G of the toric code. The exact connection with minimum distance is given in Proposition 2.1 of [13]. The determinants of the matrices that arise here can be viewed as multivariable analogs of the *Vandermonde determinants* you may have encountered in linear algebra (or possibly in a course on differential equations).

An $n \times n$ one-variable Vandermonde has the form

$$V(a_1, a_2, \ldots, a_n) = \det \begin{pmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_n \\ a_1^2 & a_2^2 & \cdots & a_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{n-1} & a_2^{n-1} & \cdots & a_n^{n-1} \end{pmatrix}$$

for variables (or elements of a field $\mathbb{F}$) $a_1, a_2, \ldots, a_n$. In our discussion of BCH codes, we have seen that

$$V(a_1, a_2, \ldots, a_n) = \prod_{1 \le i < j \le n} (a_j - a_i)$$

It follows that the Vandermonde matrix for $a_1, a_2, \ldots, a_n \in \mathbb{F}$ is invertible if and only if the a_i are *distinct elements of* $\mathbb{F}$.

The multivariable Vandermonde determinants for toric codes are determined by the exponent vectors $e \in P \cap \mathbb{Z}^m$ and subsets $S \subset (\mathbb{F}_q^*)^m$ with $|S| = |P \cap \mathbb{Z}^m|$. Here is a simple example. Say P is the convex hull of $(0,0), (1,0), (1,1), (2,1)$ (P is a

parallelogram in the plane). If we take a subset of $(\mathbb{F}_q^*)^2$ of the (special) form

$$S = \{(a,b),(c,b),(c,d),(e,d)\},$$

then the corresponding Vandermonde determinant is

$$V(P;S) = \det \begin{pmatrix} 1 & 1 & 1 & 1 \\ a & c & c & e \\ ab & cb & cd & ed \\ a^2b & c^2b & c^2d & e^2d \end{pmatrix} = \pm c(d-b)(a-c)(e-c)(de-ab).$$

(Look at this closely and try to understand the pattern.) From this we can establish conditions ensuring that $V(P;S) \neq 0$ and then determine the minimum distance of $C_P(\mathbb{F}_q)$.

Not every subset S with $|S| = 4$ has exactly the form above. These S are the ones where the Vandermonde $V(P;S)$ *factors in a nice way.* The first step of the work in [13] was devoted to determining suitable nice S where the Vandermonde $V(P;S)$ for P rectangular solids or simplices factored in a similar way. *In intuitive terms, the lesson there was that the nice S looked a lot like the sets of lattice points in the polytope P itself(!)* Then, the second step was to argue that there were enough of the nice S to yield the desired statements about the minimum distance for $C_P(\mathbb{F}_q)$.

The main goal of this project would be to try to extend the methods and results from [13] to more general classes of polytopes than the rectangular solids and special simplices studied there. For instance, good places to start would be to consider:

(1) *more general families of simplices* – e.g. the family of simplices of the form $\text{conv}(0, \ell e_1, 2\ell e_2, 3\ell e_3)$ $\ell \geq 1$ in $\mathbb{R}^3$.
(2) the polytope $\text{conv}(e_1, e_2, 2e_1+2e_2)$ from [17]. Can you analyze the minimum distance for these codes this way?
(3) *zonotopes* P as defined before – that is zonotopes in $\mathbb{R}^n$ with $g \geq n$ generators.
(4) *Cartesian products*, *Minkowski sums*, etc. of polytopes of these types.
(5) *other classes of polytopes?* – Be aware that this method could be difficult for completely general P, however. It is not clear *a priori* that *any* nice S exist where $V(P;S)$ factors as in the example above, or that there would be enough of them to get a good bound on the minimum distance.

Project 5 – Generalized Hamming weights of toric codes. One question that, to my knowledge, has not been addressed at all in previous work is the problem of determining the *generalized Hamming weights* of toric codes. The generalized Hamming weights of a linear $[n,k,d]$ code C are a sequence of integers

$$d_1(C) \leq d_2(C) \leq \cdots \leq d_k(C) \leq n$$

defined as follows. If $D \subset C$ is a subcode of dimension r, $1 \leq r \leq k$, we let the *support* of D, denoted $\text{supp}(D)$, be the set of all i, $1 \leq i \leq n$, such that some $d = (d_1, \ldots, d_n) \in D$ has $d_i \neq 0$. Stated another way, in a generator matrix for D, the nonzero columns will correspond to the elements of $\text{supp}(D)$. Then the rth Hamming weight of C is defined to be

$$d_r(C) = \min\{|\text{supp}(D)| \mid D \subset C \text{ a subcode with } \dim(D) = r\}.$$

Note that in the case $r = 1$, a subcode of dimension 1 just consists of the multiples of some nonzero vector. Hence $d_1(C) = d$, the ordinary minimum distance. The generalized Hamming weights give further information about the performance of codes in certain applications, and form an important piece in understanding the complete structure of code.

See section 7.10 in [8] for further properties of the generalized Hamming weights and some examples.

Some good questions to consider here would be:

(1) First compute lots of examples with small q, k – as far as I know, this is completely unexplored territory! Are there efficient ways to compute d_r (for toric codes, say, given the functionality in Magma's coding theory package)?
(2) How does the structure of the polytope P and its subpolytopes influence the values of the $d_r(C_P(\mathbb{F}_q))$? In particular, can you find bounds for the $d_r(C_P(\mathbb{F}_q))$ similar in spirit to the bounds from [13] or [12] and [17]?
(3) Are there families of polytopes for which some $d_r(C_P(\mathbb{F}_q))$ attain the generalized Singleton bound

$$d_r(C) \leq n - k + r$$

from [8], Theorem 7.10.6?

2. List Decoding Algorithms

General Background. Say C is a Reed-Solomon code over $\mathbb{F}_q$ with $d = 2t + 1$. In the seminar, we discussed the *key equation* for decoding Reed-Solomon codes,

$$\Lambda(x)\Sigma(x) \equiv \Omega(x) \bmod x^{2t},$$

where $\Sigma(x)$ is the *syndrome polynomial*, $\Lambda(x)$ is the *error locator polynomial*, and $\Omega(x)$ is the *error evaluator polynomial*. The syndrome polynomial is determined by the error in the received word and $\Lambda(x)$ and $\Omega(x)$ are the "unknowns." The Berlekamp-Massey algorithm and the Euclidean algorithm (Sugiyama) decoders work by finding a solution $(\Lambda(x), \Omega(x))$ of the key equation, in which $\deg \Lambda(x) \leq t$ and $\deg \Sigma(x) \leq \deg \Lambda(x) - 1$. These algebraic decoding algorithms both return the *unique* codeword c closest to the received word r when $d(r, c) \leq t$, but fail when the closest codeword(s) are farther away from r, or when there is more than one codeword at minimum distance from r.

For the past 10 years or so, since work of M. Sudan and the Ph.D. thesis of V. Guruswami, an alternative approach to decoding for Reed-Solomon and related codes has been receiving a lot of attention. Sudan's work, and later extensions due to Guruswami and Sudan deals with *list decoding algorithms*, in which the output is not just a single codeword, but a list of all codewords c satisfying $d(c, r) \leq \tau$ for some $\tau \geq t$. The integer τ is called the *decoding radius*.

Section 7.6 of [14] has a very good discussion of the Guruswami-Sudan method (much clearer than the original papers!) The basic outline is as follows. We use the description of the Reed-Solomon codewords as evaluations of polynomials of degree $\leq k - 1$ at the nonzero elements

$$(x_1, \ldots, x_n) = (\alpha^0, \ldots, \alpha^{q-2})$$

of $\mathbb{F}_q$ (or possibly some subset). Hence each codeword corresponds to a set of points in $\mathbb{F}_q^2$:

$$\{(x_i, f(x_i)) \mid 1 \leq i \leq n\}$$

for some $f(x)$ of degree $\leq k-1$. Similarly, the received word can be viewed as the set

$$\{(x_i, r_i) \mid 1 \leq i \leq n\}.$$

The decoding algorithm proceeds in two steps:

(1) **Interpolation** – First, a two-variable polynomial $Q(x, y)$ is computed to interpolate the received word. That is, $Q(x, y)$ is chosen to satisfy $Q(x_i, r_i) = 0$ for all $1 \leq i \leq n$ (with an associated *multiplicity* m for all of the points). Q is determined to be *minimal* with respect to a certain weighted degree ordering on monomials (that, in effect, forces the degree in y to be as small as possible).
(2) **Factorization** – Under the multiplicity assumption, any polynomial $Q(x, y)$ as in the first step must factor as

$$Q(x, y) = (y - p_1(x)) \cdots (y - p_L(x))R(x)$$

with $\deg p_i(x) \leq k-1$.
(3) Once the factorization is found, each $y - p_i(x)$ corresponds to a codeword c_i and the algorithm returns the *list of codewords*

$$\{c_1, \ldots, c_L\}.$$

A number of different approaches to both of the first two steps have been developed, and the bibliography of [14] contains all of the work up to about 2006.

Project 6 – When do lists of a fixed size L suffice? Begin by fixing a particular field $\mathbb{F}_q$ and consider Reed-Solomon codes over that field (say the "full codes" with $n = q-1$). If we take a particular Reed-Solomon code and a particular multiplicity m used to produce the interpolating polynomial $Q(x, y)$, then the corresponding achievable decoding radius is denoted by t_m in [14, pp. 332-333]. It is of practical interest, of course, to determine how large lists will be necessary when we consider all possible errors within the decoding radius for that m. So we can also study L_m = maximal degree in y of any $Q(x, y)$ obtained for errors of weight $\leq t_m$. For this project you would do the following:

(1) Begin by studying Section 7.6 of [14] carefully to learn the details of the Guruswami-Sudan list decoding method and understand the theory of 2-variable polynomials and monomial weight orders involved.
(2) Think about questions like this: If we decide we always want to use a certain multiplicity m and the associated t_m, what are the ranges of k for which lists of size L *always suffice*? For instance, for $m = 1$, and decoding radius t_1, which dimensions k give codes that can be decoded with lists of size (at most) $L = 2$ or $L = 3$, etc.? (The answer to this is known, but you should try to figure it out for yourself before going to sources to find an answer.)
(3) On the other hand, if we decide we can only use lists of size $\leq L$ for some fixed L, what is the largest decoding radius that can be achieved (for a given k first, then as a function of k)?

(4) Fixing n and the field $\mathbb{F}_q$ again, given k, what multiplicity $m = m(k)$ first attains the maximum list decoding radius $n - K_\infty$ (using notation from [14]) for that k? For instance, if $q = 32$, $n = 31$, then $m(k)$ as a function of k is plotted in Figure 2. Write some Maple procedures to generate the

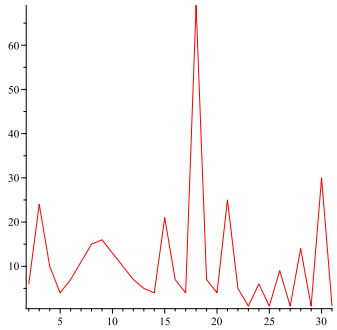

FIGURE 2. $m(k)$ as a function of k for Reed-Solomon codes over $\mathbb{F}_{2^5}$

analogous graphs for any n. Try them to reproduce this graph for $n = 31$ and generate some data to look at for other n (be sure to include cases with $n < q - 1$ over $\mathbb{F}_q$). Is there always only one k for which $m(k)$ reaches its maximum? If so, how does that k depend on q, n?

(5) You might also think of implementing the algorithms presented in [14] in Maple to automate working out larger examples if you want to explore the actual Guruswami-Sudan decoding process.

Project 7 – A new method for the Interpolation step. In the recent article [11], Lee and O'Sullivan have proposed a way to think about the Interpolation step of the Guruswami-Sudan list decoding algorithm in terms of computing a Gröbner basis for a certain module of a polynomial ring with respect to a particular monomial order. The method they use, though, does not make use of the fact that they already know a Gröbner basis of the module with respect to a different monomial order. In a situation like this, there is a *basis conversion method* called the FGLM algorithm that could be used instead.

If one wanted to make full use of the fact that the original set of generators for the module $I_{v,m,l}$ are actually also a Gröbner basis, but with respect to a different order on $\mathbb{F}[x,y]_l$ (the order induced from the lexicographic order on $\mathbb{F}[x,y]$ with $y > x$). On Example 9 from [11], for instance, it seems looks that an "optimized" FGLM (making use of the shape of the lex "footprint") would need very few actual computations to find the minimal element $Q(x,y)$.

The idea is to list the monomials in increasing $>_{k-1}$ (here $>_2$) order, compute the remainders (normal forms) with respect to the original lex Gröbner basis. We can stop the first time we hit a linear dependence between the remainders – that will give the $>_2$-minimal element of the module. Now in this example, $\dim_{\mathbb{F}}(k[x,y]_3/M) = 18$, with a "footprint" basis

$$\{1, x, x^2, \ldots, x^{11}, y, xy, x^2y, \ldots, x^5y\}$$

Note the original lex basis $\{g_0, g_1, g_2, g_3\}$ is not reduced because of the terms in g_3 divisible by leading terms of g_1 and g_2. To save some work later, we could start by replacing g_3 by $g_3 - (5x^4 + \cdots)g_2$, then reduce the result with g_1 and g_0, so the new g_3 has form $g_3 = y^3 + \cdots$, where the $\cdots$ represent terms in footprint – three polynomial divisions in $\mathbb{F}[x]$.

Now we start the FGLM procedure, listing the monomials in increasing $>_2$ order starting from 1. For the first batch,

$$\{1, x, x^2, y, x^3, xy, x^4, x^2y\}$$

no remainders are necessary since these are in the footprint already. The next one, y^2 is the leading term of the module element g_2, but all the other terms are in the footprint, and this is clearly independent of the ones we already know because of the x^8 (and the x^4y). Then we continue:

- x^5 (in footprint and different $>_2$ leading term from any remainder we have seen so far, so OK)
- x^3y (ditto)
- xy^2 (ditto – x^9 and x^5y terms)
- y^3 (use modified g_3, get a x^{11} term so still can see linear dependence without a lot of calculation)
- x^7 (ditto – leading term x^7 is new)
- x^5y (here you do need to check that this and previous remainders are linearly independent, but they are essentially taking an echelon form matrix where most rows have only one non-zero entry, adding one more row and reducing it.)
- x^3y^2 (the 19th monomial considered, and you get the expected linear combination yielding the correct $Q(x, y)$).

In any case, the nice thing that happened here was that so many of the monomials you need to process are already in the lex footprint, so you don't need to do a lot of computations.

(1) This project is slightly different from some of the others in that it draws on a lot of background we have not discussed at all in the seminar. However, Gröbner bases are a quite important algebraic tool in contemporary coding theory, so learning about this extra topic would definitely be worthwhile if you wanted to continue to study this field. Good sources are [2, Chapter 2] and [3, Chapter 2].
(2) The project would be to first learn the relevant material about Gröbner bases and the FGLM basis conversion algorithm to understand the outline computation above.
(3) Then, study whether this proposed different method is really an improvement over the method proposed in [11].
(4) Work out some additional examples and compare the amount of computation needed for each method.
(5) Try to prove a general pattern!

References

[1] P. Cameron, *Polynomial aspects of codes matroids and permutation groups*, notes available online at `http://www.maths.qmw.ac.uk/~pjc/csgnotes/cmpgpoly.pdf`

[2] D. Cox, J. Little, D. O'Shea, *Ideals, Varieties, and Algorithms*, 3rd edition, Springer, New York, 2008.
[3] D. Cox, J. Little, D. O'Shea, *Using Algebraic Geometry*, 2nd edition, Springer, New York, 2006.
[4] V. Diaz, C. Guevara, M. Vath, *Codes from n-Dimensional Polyhedra and n-Dimensional Cyclic Codes*, Proceedings of SIMU summer institute, 2001.
[5] M. Grassl, *Code Tables: Bounds on the parameters of various types of codes*, `http://www.codetables.de`.
[6] J. Hansen, *Toric surfaces and error-correcting codes*, in Coding theory, cryptography and related areas (Guanajuato, 1998), 132–142, Springer, Berlin, 2000.
[7] J. Hansen, *Toric varieties Hirzebruch surfaces and error-correcting codes*, Appl. Algebra Engrg. Comm. Comput. **13** (2002), 289–300.
[8] W. Huffman and V. Pless, *Fundamentals of Error-Correcting Codes*, Cambridge University Press, Cambridge, 2003.
[9] D. Joyner, *Toric codes over finite fields*, Appl. Algebra Engrg. Comm. Comput. **15** (2004), 63–79.
[10] D. Joyner, `http://www.usna.edu/Users/math/wdj/magma/magmastuff.html`.
[11] K. Lee, M. O'Sullivan, *List decoding of Reed-Solomon codes from a Gröbner basis perspective*, J. Symb. Comp. **43** (2008), 645–658.
[12] J. Little, H. Schenck, *Toric surface codes and Minkowski sums*, SIAM J. Discrete Math **20** (2006), 999-1014.
[13] J. Little, R. Schwarz *On toric codes and multivariate Vandermonde matrices*, Appl. Alg. Engrg. Comm. Comput. **18** (2007), 349-367.
[14] T. Moon, *Error Correction Coding*, Wiley-Interscience, Hoboken, 2005.
[15] D. Ruano, *On the parameters of r-dimensional toric codes*, Finite Fields Appl. **13** (2007), 962-976.
[16] D. Ruano, *On the structure of generalized toric codes*, to appear in J. Symb, Comp,, `arXiv:cs.IT/0611010`.
[17] I. Soprunov, J. Soprunov, *Toric surface codes and Minkowski length of polygons*, SIAM J. Discrete Math **23** (2009), 384-400.
[18] G. Ziegler, *Lectures on Polytopes*, Springer Verlag, Berlin, 1995.