Neville Fogarty

Research Statement

---

1 Algebraic Coding Theory

When we transmit data over a noisy channel (e.g., satellite) to a receiver, we run the risk of our data becoming corrupted. Fixing the channel is not typically an option, so instead we must make our data noise-proof. A naive method of doing this is simply sending the same data multiple times. However, the benefit of the redundancy created may be offset by the cost of sending a lengthy message more than once in full. Indeed in some cases, such as transmissions through space via satellite, sending a message more than once may be entirely impractical. The study of algebraic coding theory is about balancing the added redundancy with its costs and finding new ways to efficiently encode and decode information.

Since the symbols of the to-be-transmitted data come from a finite set, algebraic coding theory falls, in large part, within linear algebra over finite fields. (Particular codes may draw from other areas, such as elliptic curves and polynomial rings, as well.) Let $\mathbb{F}$ be a finite field of size q. At this point, one may simply think of $\mathbb{F}_2 = \{0, 1\}$, where we add and multiply modulo 2. A *linear code* $\mathcal{C}$ is a k-dimensional subspace of the vector space $\mathbb{F}^n$. Each vector $\mathbf{c} \in \mathcal{C}$ is a codeword; each codeword represents a known message. When we send the codeword $\mathbf{c}$ over a noisy channel, it picks up an error, $\mathbf{e} \in \mathbb{F}^n$. (Note that the error $\mathbf{e}$ could be the zero vector.) The receiver gets the vector $\mathbf{c} + \mathbf{e}$. Armed with the knowledge of which codewords are contained within the subspace $\mathcal{C}$, the receiver can often recognize codewords that were affected by channel noise (provided $\mathbf{e} \notin \mathcal{C}$). This is called error detection.

Moreover, when we create codes wisely, the receiver can not only detect erroneous codewords, but also correct them by changing them to the 'closest' allowable codewords. The *Hamming distance*, $\text{dist}(\mathbf{c}, \mathbf{c}')$, between two vectors $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$ is defined as the number of entries in which the two vectors differ. For example, $\text{dist}((1, 0, 1, 1), (0, 1, 1, 0)) = 3$, as the vectors differ in the first, second, and fourth entries. Then the *minimum distance* of code $\mathcal{C}$ is the smallest distance between any two distinct codewords. In order to correct as many errors as possible, we want to employ a code with a large distance. If a code has distance d, then the receiver will correctly decode a received vector $\mathbf{c} + \mathbf{e}$ to $\mathbf{c}$, provided that $\mathbf{e}$ has at most $\lfloor (d-1)/2 \rfloor$ non-zero entries. Otherwise, the receiver will not always be able to decode error-laden vectors correctly.

2 Cyclic Codes

A *cyclic code* is a code in which cyclic shifts of codewords produce other codewords. That is, if $(c_0, c_1, ..., c_{n-2}, c_{n-1}) \in \mathcal{C}$, then $(c_{n-1}, c_0, c_1, ..., c_{n-2}) \in \mathcal{C}$ as well. Introduced by Prange in 1957 [11], cyclic codes are known to have nice error-correcting properties if suitably chosen. We obtain a cyclic code by considering the ideals of the quotient ring $\mathbb{F}[x]/(x^n - 1)$. Since $\mathbb{F}[x]$ is a principal ideal domain, each ideal in the quotient ring is generated by a single element. Naturally, we call the monic polynomial (coset representative) of minimum degree in the ideal the *generator polynomial*, but other polynomials can generate the same code. We get the codeword vectors by reading off the coefficients of the polynomials in the given ideal. Notice that multiplying a polynomial in our ideal by x is the same as applying a right cyclic shift to the associated codeword.

We say that $e \in \mathbb{F}[x]/(x^n - 1)$ is an *idempotent* if $e^2 = e$. Because of the presence of zero divisors in the ring (provided $x^n - 1$ is not irreducible), there are indeed idempotents other than 0

and 1. Moreover, if a code $\mathcal{C}$ is generated by e, we say that e is a *generating idempotent* of $\mathcal{C}$. Provided that n is relatively prime to q, the size of the field $\mathbb{F}$, one can show that $\mathcal{C}$ contains a unique generating idempotent. (See, for example, [7].)

When we want to create a code, finding a generator polynomial amounts to factorizing $x^n - 1$, which can be difficult. On the other hand, idempotents can easily be found with the aid of cyclotomic cosets. Thus it is beneficial to understand generating idempotents for cyclic codes.

3 Skew-constacyclic Codes

In 2007, Boucher, Geiselmann, and Ulmer [1] generalized cyclic codes to *skew-constacyclic codes*. Instead of considering a standard polynomial ring $\mathbb{F}[x]$, they looked at the *skew-polynomial ring* $\mathcal{R} := \mathbb{F}[x;\theta]$, where θ is an automorphism of $\mathbb{F}$. The skew-polynomial ring was introduced by Ore in 1933 [10]. While addition of polynomials in $\mathcal{R}$ is identical to addition in the standard polynomial ring, multiplication is different. Critically, $xa = \theta(a)x$ for all $a \in \mathbb{F}$. If θ is the identity, we are back in the standard polynomial ring, but otherwise $\mathcal{R}$ is noncommutative. Despite the noncommutativity, one can still perform left and right division with remainder.

Skew-constacyclic codes are certain subspaces of the quotient $\mathcal{R}/^{\bullet}(x^n - a)$, where $^{\bullet}(x^n - a)$ is the left ideal generated by $x^n - a$, with $a \in \mathbb{F}^*$. This clearly generalizes cyclic codes. Different from the classical cyclic case, the quotient $\mathcal{R}/^{\bullet}(x^n - a)$ is not a ring in general, but rather a left $\mathcal{R}$-module. By definition, skew-constacyclic codes are the left $\mathcal{R}$-submodules of that quotient module. Since $\mathcal{R}/^{\bullet}(x^n - a)$ is naturally isomorphic as a left $\mathbb{F}$-vector space to $\mathbb{F}^n$, we again obtain linear codes as described earlier. Similar to the classical cyclic case, one obtains that each left-submodule is generated by the coset of a right divisor of $x^n - a$. Thus one has an analogous concept of a generator polynomial. To capture data about the submodule, we typically call these codes (θ, a)-constacyclic codes. Throughout this statement, we keep θ fixed and call the codes just described *a-constacyclic codes*.

Recall that error-correcting quality relies on having a large minimum distances. In [1] and [4], skew-constacyclic codes were presented for particular parameters (such as code length n) that improved upon the largest known minimum distances of other codes. This strongly suggests that there is indeed potential in this burgeoning area of coding theory.

4 Duals of a-constacyclic Codes

The *dual code* of a code is the space orthogonal to the code when viewed as a subspace of $\mathbb{F}^n$. More precisely, the dual code of a code $\mathcal{C} \in \mathbb{F}^n$ is denoted by $\mathcal{C}^{\perp} := \{\mathbf{x} \mid \mathbf{x} \cdot \mathbf{c} = \mathbf{0} \text{ for all } \mathbf{c} \in \mathcal{C}\}$. In [2], Boucher and Ulmer showed the following theorem:

Theorem 1. *If $\mathcal{C}$ is an a-constacyclic code, then $\mathcal{C}^{\perp}$ is an a^{-1}-constacyclic code. Moreover, if $\mathcal{C}$ has generator polynomial g, where $x^n - a = hg$, then $\mathcal{C}^{\perp}$ has generator polynomial $\hat{h}$, given as $\hat{h} := \sum_{i=0}^{k} \theta^{i-n}(h_{k-i})x^i$, where $h = \sum_{i=0}^{k} h_i x^i$. Thus $\hat{h}$ is a right divisor of $x^n - a^{-1}$.*

This generalizes a well-known result from the theory of cyclic codes. The proof in [2] relies primarily on computations inside the code viewed as a vector subspace. In [5], we reformulated the work of [2] to instead rely on the properties of the skew-polynomial ring. In doing so, we introduced a tool called a *(skew-generalized) circulant*, a matrix in $\mathbb{F}^{n\times n}$ which captures the effects of multiplying a skew polynomial g by powers of x on the left. Precisely, the ith row of the circulant is the list of left coefficients of the polynomial $x^{i-1}g$ modulo $^{\bullet}(x^n - a)$. This matrix clearly depends on the constant a, so we denote it by $M_a(g)$.

Using properties of the skew-polynomial ring, we have the following theorem regarding circulants, which summarizes some of our work in [5].

Theorem 2. *Let $x^n - a = hg$, where g has constant term 1.*
(a) If g has degree $n-k$, then the first k rows of $M_a(g)$ form a basis of the rowspace of $M_a(g)$, which in turn is the code generated by g.
(b) $M_a(fg) = M_a(f)M_a(g)$ for all $f \in \mathcal{R}$.
(c) $M_a(g)^{\mathsf{T}} = M_{a^{-1}}(g^{\#})$ for some polynomial $g^{\#}$ (whose definition we omit here).
(d) $M_a(g)M_{a^{-1}}(\hat{h})^{\mathsf{T}} = 0$, and the ranks of the two matrices add up to n.

From this, we immediately recover the results from [2] presented in Theorem 1. It is worth pointing out that Theorem 2(c) is not true if g is not a right divisor of $x^n - a$; the transpose of a circulant is not generally another circulant.

5 Generating Idempotents in a-constacyclic Codes

Recall that in the classical cyclic case, finding generating idempotents is easier than factorizing $x^n - a$ to find generator polynomials if n is large. Finding factorizations of $x^n - a$ in the skew-polynomial ring $\mathcal{R}$ is even harder, as $\mathcal{R}$ is not a unique factorization domain. To this end, we want to generalize results from cyclic codes to the skew-constacyclic case.

Since $\mathcal{R}/{}^{\bullet}(x^n - a)$ is in general a module and *not* a ring, we must modify our definition of an idempotent. We now say that a skew-polynomial $e \in \mathbb{F}[x;\theta]$ is an *idempotent modulo* ${}^{\bullet}(x^n - a)$ if $e^2 - e \in {}^{\bullet}(x^n - a)$, and that e is a *generating idempotent of* ${}^{\bullet}(\overline{g})$ if, additionally, ${}^{\bullet}(\overline{g}) = {}^{\bullet}(\overline{e})$.

In [6], Gao, Shen, and Fu demonstrate a method for finding idempotents modulo ${}^{\bullet}(x^n - a)$ when $a = 1$ and n is a multiple of $|\theta|$. In my dissertation, I improve their method to work for any constant $a \in \mathbb{F}^*$:

Theorem 3. *Let $x^n - a$ be central, and let g be a central right divisor of $x^n - a$ generating an a-constacyclic code $\mathcal{C}$. Then $\mathcal{C}$ has a unique generating idempotent of degree less than n.*

When $x^n - a$ is not central, finding idempotents becomes significantly more complicated. In the cyclic case, the restriction to $\gcd(n, q) = 1$ is necessary. In the θ-constacyclic case, we conjecture, based on an abundance of examples, that obtaining generating idempotents will require similar restrictions.

Conjecture 4. *Suppose $\gcd(n, q) = \gcd(n, |\theta|) = 1$ and $x^n - a = hg$. Suppose further that g has constant term 1. Then:*
(a) The greatest common right divisor of h and g is 1.
(b) Let $1 = uh + vg$ with $\deg(u) < \deg(g)$ and $\deg(v) < \deg(h)$. Then $vg = gv$.

Note that performing the rescaling of our generator polynomial g so that it has constant term 1 is not difficult, but unlike in Theorem 2, here it is a necessary component of our conjecture. Assuming Conjecture 4, we can produce generating idempotents for skew-constacyclic codes:

Theorem 5. *Let $\mathcal{C}$ be an a-constacyclic code generated by g, where $x^n - a = hg$ with $1 = uh + vg$ and $vg = gv$ for some $u, v \in \mathcal{R}$. Then vg is a generating idempotent of $\mathcal{C}$.*

This leads to a broad class of codes with generating idempotents. These idempotents in turn can be used to generalize well-known results of idempotents of sums and intersections of classical cyclic codes.

6 Future Goals

I naturally hope to resolve Conjecture 4, which would imply a large class of codes with generating idempotents. In contrast to the cyclic case, a-constacyclic codes can have multiple generating idempotents. Under what circumstances does a a-constacyclic code have a *unique* generating idempotent?

There are some clear connections between a-constacyclic codes and generalize Vandermonde matrices. Leroy and Lam defined a generalized Vandermonde for this purpose in [8]. In [6], Gao, Shen, and Fu gave a lower bound on the Hamming distance of an a-constacyclic code $\mathcal{C}$ by using the generalized Vandermonde matrix. They required, as in the cyclic case, that the generator polynomial of $\mathcal{C}$ has (right) roots that are consecutive powers of a primitive element of a particular extension of $\mathbb{F}$. Boulagouaz and Leroy [3] examined how the generator polynomial g of an a-constacyclic code behaves when it is a Wedderburn polynomial. The right roots of g can then be examined via a Vandermonde matrix. In [9], Liu, Manganiello, and Kschischang used the generalized Vandermonde matrix to find minimal degree skew-polynomials that vanish over a set of field elements. We should be able to generalize and combine the results so far. This could more easily allow us to obtain information on the distances of a-constacyclic codes.

7 Undergraduate Research

I am excited to share my research area with undergraduate students. My own research opportunities as an undergraduate — an independent study, a Research Experience for Undergraduates, and a National Security Agency internship — helped inspire me to pursue a career in mathematics. I want to inspire future mathematicians in the same way by working with them on questions from my own area. Coding theory lends itself easily to undergraduate research. The concept of a vector space over a finite field should be clear after an undergraduate course in abstract algebra; the leap from there to coding theory is minimal. And as an area with obvious real world applications, coding theory has appeal to both pure and applied mathematicians. My own current research requires a grasp of noncommutative rings, which could lend itself to an interesting independent study for an upper-division student with a strong interest in algebra. In particular, much of my own work has involved the use of the computer algebra system Maple to formulate conjectures and generate examples. This can provide an opportunity for students interested in programming and contributing to research with an abstract approach.

In addition, one can take a more concrete view of codes by considering them through the lens of graph theory. I am interested in taking this perspective as well and working with undergraduates to visualize codes. There are a number of ways of applying graph theory to coding theory.

For example, a *Hamming graph* $H_m(n, d)$ is a graph with vertices corresponding to vectors in $(\mathbb{Z}/m\mathbb{Z})^n$. Two vertices u, v are connected by an edge if and only if $\text{dist}(u, v) \geq d$. The Hamming graph encapsulates what it means for codewords to be sufficiently far apart. Thus finding the largest code (potentially non-linear) of a particular distance d (with n and m fixed) is equivalent to finding the largest complete subgraph of $H_m(n, d)$.

We can also view a code as a *trellis*, which is a specific type of directed graph. Vertices of a trellis are partitioned into $n + 1$ sets with an ordering, and edges are directed from the ith set to the $(i + 1)$st set. Each edge is labelled with a field element. The code represented by the trellis consists exactly of the vectors that can be read across any path of length n.

Another class of codes, *low-density parity-check codes*, are constructed via bipartite graphs. One partite set of vertices corresponds to entries in a potential codeword, while the other partite set serves as a collection of parity checks to determine if the codeword is valid.

We see that different types of codes and different questions lead to their own graphical interpretations. Each of these perspectives is a readily available approach to coding theory for interested undergraduates. Combining the idea of a graphical representation of a code with my own area of research, we can ask: is there a natural way to interpret or view a skew-constacyclic code as a graph?

References

[1] D. Boucher, W. Geiselmann, and F. Ulmer. Skew-cyclic codes. *Applicable Algebra in Engineering, Communication and Computing*, 18(4):379–389, 2007.

[2] D. Boucher and F. Ulmer. A note on the dual codes of module skew codes. In *Cryptography and coding*, volume 7089 of *Lecture Notes in Comput. Sci.*, pages 230–243. Springer, Heidelberg, 2011.

[3] M. Boulagouaz and A. Leroy. (σ, δ)-codes. *Advances in Mathematics of Communications*, 7(4):463–474, 2013.

[4] L. Chaussade, P. Loidreau, and F. Ulmer. Skew codes of prescribed distance or rank. *Designs, Codes and Cryptography*, 50(3):267–284, 2009.

[5] N. Fogarty and H. Gluesing-Luerssen. A circulant approach to skew-constacyclic codes. *Finite Fields and Their Applications*, 35(0):92 – 114, 2015.

[6] J. Gao, L. Shen, and F. Fu. Skew generalized quasi-cyclic codes over finite fields. arXiv:1309.1621, 2013. Preprint.

[7] W. C. Huffman and V. Pless. *Fundamentals of error-correcting codes*. Cambridge Univ. Press, 2003.

[8] T. Lam and A. Leroy. Vandermonde and wronskian matrices over division rings. *Journal of Algebra*, 119(2):308 – 336, 1988.

[9] S. Liu, F. Manganiello, and F. Kschischang. Kötter interpolation in skew polynomial rings. *Designs, Codes, and Cryptography*, 72(3):593–608, 2014.

[10] O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics*, 34(3):pp. 480–508, 1933.

[11] E. Prange. *Cyclic error-correcting codes in two symbols*. Electronics Research Directorate, Air Force Cambridge Research Center, September 1957. No. AFCRC-TN-57-103. ASTIA Document No. AD133749.