Math 75 notes, Lecture 19 outline

P. Pollack and C. Pomerance

References below are to Pretzel's *Error-correcting codes and finite fields*:

- We reviewed the characterization of $\mathrm{BCH}(k,t)$ derived in Lecture 18: Let α be a generator for $\mathbb{F}_{2^k}^{\times}$. Identify a word $u = (u_{n-1}, \ldots, u_1, u_0) \in \mathbb{F}_2^n$ with the polynomial $u(x) = u_{n-1}x^{n-1} + \cdots + u_1 x + u_0$. (This is a one-to-one correspondence between the words u of length n and the polynomials of degree $\leq n-1$ over $\mathbb{F}_2$, together with the zero polynomial.) Then $u(x)$ is a code word exactly when $g(x)$ divides $u(x)$, where $g(x)$ is the product of the distinct minimal polynomials of $\alpha, \alpha^2, \ldots, \alpha^{2t}$.

- We reviewed one consequence for the rank m of $\mathrm{BCH}(k,t)$: we have $m = n - \deg g(x)$.

- We reviewed that encoding can be done easily, and amounts to multiplying by $g(x)$. (This is only one encoding scheme, called *multiplicative encoding*; the book also discusses *systematic encoding*, which we didn't get to.)

- We showed that $g(x)$ divides $x^n - 1$. (Thus, in the terminology introduced below, each of the BCH codes is cyclic.) The quotient $(x^n-1)/g(x)$ was denoted $h(x)$, and called the *check polynomial* (see p. 222). We saw that $h(x)$ could be used for recognizing code words and decoding (see p. 223).

- As a generalization of the above situation, we introduced *polynomial codes*: codes C of length n where the code words are exactly the polynomials of degree $< n$ (or identically zero) divisible by a prescribed *generator polynomial* $g(x)$ of degree $< n$. We proved that the generator polynomial $g(x)$ of a polynomial code is unique up to multiplication by a nonzero element of the field, and that for any polynomial code C, we have $m = n - \deg g(x)$. (Here, as usual, m is the rank of the code). (See p. 226.)

- We defined a *cyclic* code as a polynomial code for which the generator $g(x)$ divides $x^n - 1$ (p. 228). As above, the quotient $(x^n-1)/g(x)$ is called the *check polynomial* of the code.

- We noted that for any polynomial code, encoding can be done exactly as above, by multiplying by $g(x)$. Similarly, for any cyclic code, one can recognize code words and decode using the check polynomial.

- We proved that a linear code C of dimension ≥ 1 is a polynomial code exactly when it is closed under *left shifts* of code words with first coordinate zero. (For the definition of a *left shift*, see p. 227.) We also stated, but did not have time to prove, that a code C is cyclic exactly when it is closed under all left shifts. (See the Theorems on pages 227, 229.)