Math 75 notes, Lecture 18 outline

P. Pollack and C. Pomerance

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

- We reviewed material from Lecture 17 and also Lecture 11. In particular we discussed how to find a generator element in a finite field and illustrated this for the finite field $\mathbb{F}_{2^4}$.
- The new material we covered was in Ch. 14, sections 14.1 through 14.4.
- In particular we learned how to think of a vector in $\mathbb{F}_{2^n}$ as a polynomial: the vector $s = (s_{n-1}, s_{n-2}, \ldots, s_0)$ corresponds to the polynomial $s(x) = s_{n-1}x^{n-1} + s_{n-2}x^{n-2} + \cdots + s_0$. Here, each s_i is in $\mathbb{F}_2$ so is 0 or 1.
- If $\alpha^{j(n-1)}, \alpha^{j(n-2)}, \ldots, \alpha^{j\cdot 0}$ is the jth row of the matrix $V_{k,t}$, then the dot product of this row with the column vector s is exactly $s(\alpha^j)$. Thus, a polynomial $s(x)$ corresponds to a code word if and only if $s(\alpha^j) = 0$ for $j = 1, 2, \ldots, 2t$.
- The condition $s(\beta) = 0$ for some $\beta \in \mathbb{F}_{2^k}$ occurs if and only if $s(x)$ is divisible by the minimum polynomial of β. So, let $p_j(x)$ be the minimum polynomial of α^j. Then $s(x)$ corresponds to a code word if and only if it is divisible by each $p_j(x)$ for $j \le 2t$, and this occurs if and only if $s(x)$ is divisible by $g(x)$, the least common multiple of the various $p_j(x)$.
- The polynomial $g(x)$ as just described is called the generator polynomial for the code. Say it has degree d. Then there are 2^{n-d} words of length n that correspond to multiples of $g(x)$, that is, the dimension of $\mathrm{BCH}(k,t)$ is $n - d$.
- We computed $g(x)$ for BCH(4,3) and found it has degree 10. Thus, the dimension of the code is $15 - 10 = 5$. (Compare this with the earlier result that the dimension is $\ge n - kt$, which in this case asserts that the dimension is ≥ 3.)
- We learned one way to encode words of length $n - d$. Namely write such as a polynomial $b(x)$ of degree $< n - d$ (or possibly the 0-polynomial), and multiply by $g(x)$. Then $b(x)g(x)$ is the encoded version of $b(x)$. Conversely, one can divide a received word by $g(x)$ to retrieve the real word $b(x)$ (or detect that an error was made, if the division does not go exactly).