We are all in this together

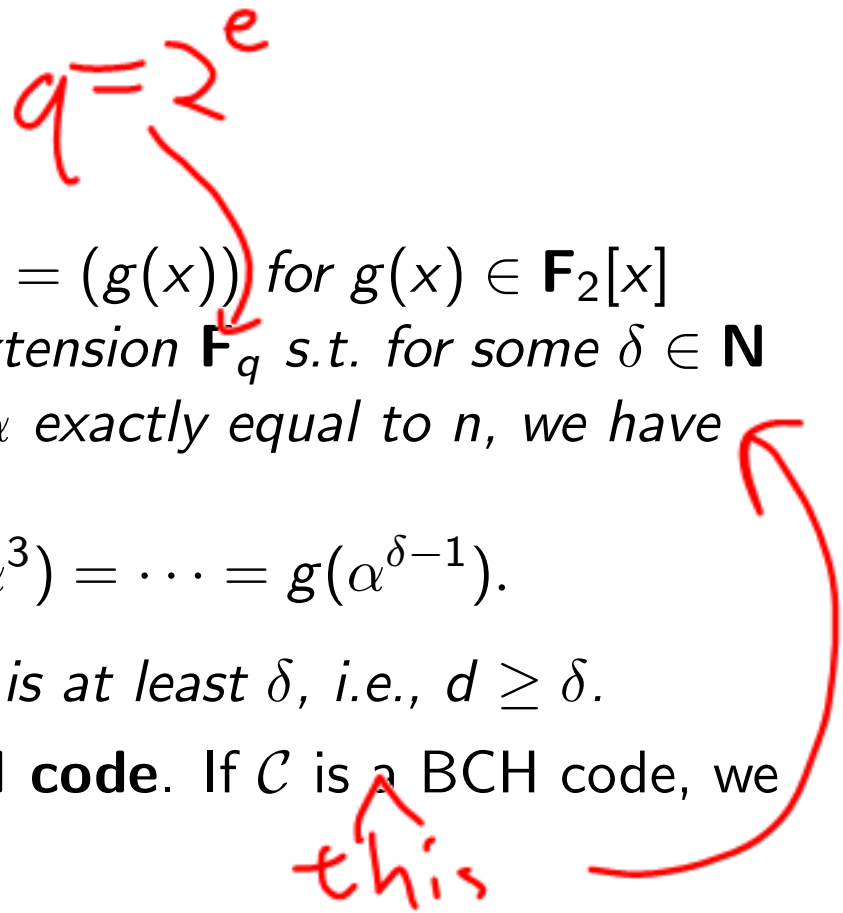
And we will get through this together.

- Use a laptop or desktop with a large screen so you can read these words clearly.
- To conserve bandwidth, please turn off your camera.
- Please mute your microphone unless I call on you.
- Please have the chat window open to ask questions.
- Reading for today: 9.2–9.4. Reading for Mon: 10.1–10.3.
- Problem session Fri 10:30–noon.
- Today's DJ: Jennifer.

PS10 or whatever else you want to catch up on

Remember two things: 1. All deadlines are flexible
2. HW is 35% of semester grade, and much of that will be just getting it done.

Last: The BCH Theorem



Theorem

$\mathcal{C}$ cyclic code of length n over $\mathbf{F}_2$, $\mathcal{C} = (g(x))$ for $g(x) \in \mathbf{F}_2[x]$ dividing $x^n - 1$. Suppose E is an extension $\mathbf{F}_q$ s.t. for some $\delta \in \mathbf{N}$ and some $\alpha \in E$ with the order of α exactly equal to n, we have that

$$0 = g(\alpha) = g(\alpha^2) = g(\alpha^3) = \cdots = g(\alpha^{\delta-1}).$$

Then the minimum distance d of $\mathcal{C}$ is at least δ, i.e., $d \geq \delta$.

A code of this form is called a **BCH code**. If $\mathcal{C}$ is a BCH code, we call δ the **designed distance** of $\mathcal{C}$.

So to get a BCH code of designed distance δ, want the smallest possible degree polynomial $g(x)$ divisible by

$$(x - \alpha)(x - \alpha^2) \ldots (x - \alpha^{\delta-1}).$$

I.e., we want the least common multiple of the minimal polynomials of $\alpha, \ldots, \alpha^{\delta-1}$.

A recipe for BCH codes

Algorithm Bose–Chaudhuri–Hocquenghem

1. Choose a finite field E, $|E| = 2^e$.
2. Choose $\alpha \in E$, and let n be the order of α. (Note that if $n = 2^e - 1$, then α is a primitive root of E.) Our code will have length n.
3. Choose a designed distance $\delta \in \mathbf{N}$. (Correct $\geq \left\lfloor \frac{\delta - 1}{2} \right\rfloor$ errs)
4. Let $g(x)$ be the least common multiple of minimal polynomials $m_1(x), \ldots, m_{\delta-1}(x)$, i.e., remove repetitions of minimal polynomials and take the resulting product.

Then $\mathcal{C}$ is the cyclic code of length n generated by $g(x)$.

If we choose $q = 2^e$ and α wisely (i.e., we get lucky), we get a code with a large dimension and a large minimal distance. Note that we don't choose n; nature chooses a "good" n for us, often $n = q - 1$.

Example: $E = \mathbf{F}_{32}$, α primitive, $\delta = 7$

$|\alpha| = 31$, so $n = 31$, $\alpha^{31} = 1$

Orbits:

(Frob) $\{\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}\}$ $\quad 32 \pmod{31}$

$[1, 2, 4, 8, 16]$

$3 \cdot 4 \pmod{31}$

$[3, 6, 12, 24, 17]$

orbit of 1, 2, 4

$3 \cdot 5 \pmod{31}$

$[5, 10, 20, 9, 18]$

We want LCM of min polynomials m_1, m_2, m_6. m_i comes from orbit of i.

so exponents all mod 31

1 2 3 4 5 6

up to 6

$= 7 - 1$

Generator:

cancels, all coeffs 0 or 1

$m_1(x) = (x-\alpha)(x-\alpha^2)(x-\alpha^4)(x-\alpha^8)(x-\alpha^{16}) \in \mathbb{F}_2[x]$

$= m_2(x) = m_4(x)$

$m_3(x) = (x-\alpha^3) \cdots (x-\alpha^{17}) = m_6(x)$

$m_5(x) = (x-\alpha^5) \cdots (x-\alpha^{18})$

Some details for [31, 21, 7] BCH code

Turns out that $\mathbf{F}_{32} = \mathbf{F}_2[\alpha]$, where $\alpha^5 = \alpha^2 + 1$, and α is primitive.

$g(x) = m_1(x)\, m_3(x)\, m_5(x)$ / deg $g = 15$

deg 5 deg 5 deg 5

dim $\mathcal{C} = 31 - 15 = 16$

min dist ≥ 7 (actually 7)

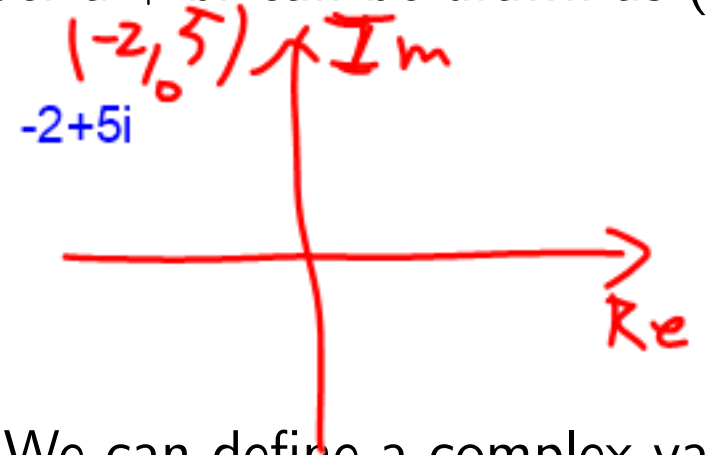
So $\mathcal{C}$ is a $[31, 16, 7]$ code

Corrects 3 errors per word

16 info bits per 31 transmitted.

A quick look at the complex plane

A complex number $a + bi$ can be drawn as (a, b) in the plane:



Fact/definition: We can define a complex-valued function $e^{i\theta}$, (θ real) satisfying the usual properties of an exponential function, such that $e^{i\theta}$ is the point on the unit circle forming the angle θ (in radians) with the positive x-axis:

$e^{i\theta} = \cos\theta + i\sin\theta$

equiv to sum formulas

Important: $e^{i\alpha}e^{i\beta} = e^{i(\alpha+\beta)}$, i.e., multiplying points on the unit circle is the same as adding their angles.

Nth roots of unity

Definition

The **natural primitive Nth root of unity** in $\mathbf{C}$ is

$$\omega_N = e^{2\pi i/N}.$$

$\omega_N^N = \left(e^{\frac{2\pi i}{N}}\right)^N = e^{2\pi i} = 1$

When N fixed, we abbreviate ω_N as ω

The important fact about ω_N is:

complex

Theorem

Let N be a positive integer, and let $\omega = \omega_N = e^{2\pi i/N}$. The zeros of the polynomial $z^N - 1$ (i.e., the solutions to $z^N = 1$) are precisely the powers $1, \omega, \omega^2, \ldots, \omega^{N-1}$ of ω.

Why $(\omega_N^k)^N - 1 = (\omega_N^N)^k - 1 = 1^k - 1 = 0$. ☺

Recap of ω_N

Let N be a positive integer, and let $\omega = \omega_N = e^{2\pi i/N}$.

1. The solutions to $z^N = 1$ are precisely the powers $1, \omega, \omega^2, \ldots, \omega^{N-1}$.
2. We have that

$$1 + \omega + \cdots + \omega^{N-1} = 0.$$

Proof of second fact on PS10, or look at following picture:

$N = 5$

Sum of powers of omega
= N*average of powers of omega
= N * "center of gravity" of regular N-gon
= N * 0 = 0.

Signals

And now for something completely different!

Definition

{0,1,2,....N-1}

Fix $N \in \mathbf{N}$. A **signal** is a function $f : \mathbf{Z}/(N) \to \mathbf{C}$, or in other words, a complex-valued function with domain $\mathbf{Z}/(N)$. Note that a signal f is defined by its N values $f(0), \ldots, f(N-1) \in \mathbf{C}$, so we sometimes represent a signal f as the vector $\begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}$.

Idea of domain being $\mathbf{Z}/(N)$ is signal periodic with period N

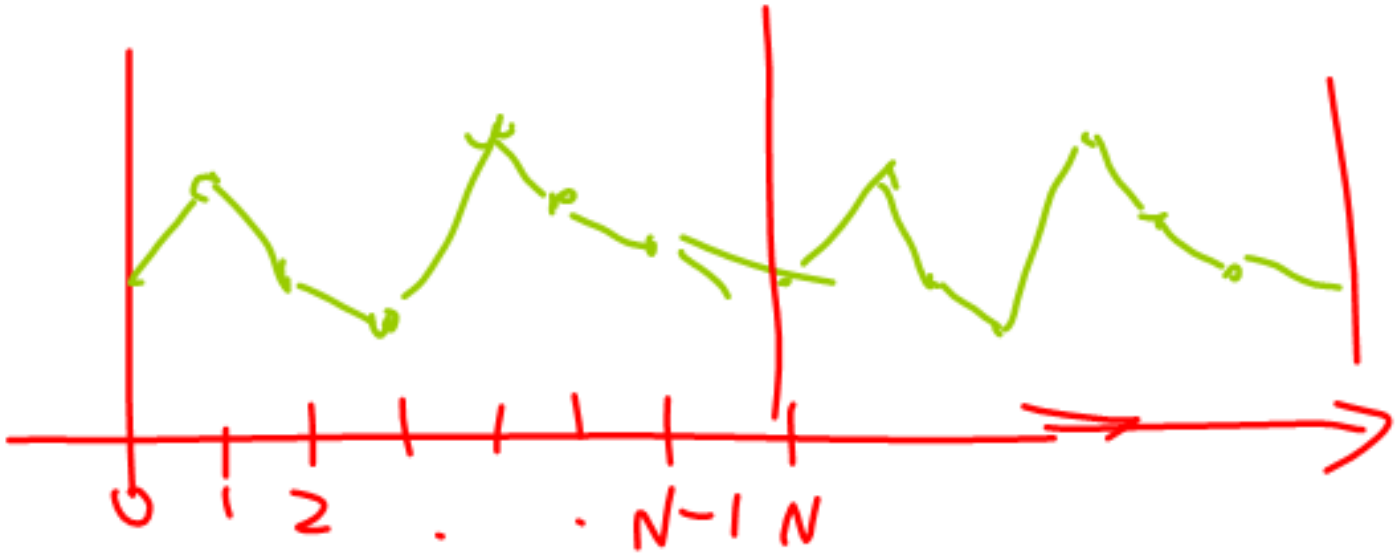
Example: Fix $\omega = \omega_N = e^{2\pi i/N}$. Then $\begin{bmatrix} 1 \\ \omega^k \\ \ldots \\ \omega^{(N-1)k} \end{bmatrix}$ is a signal $(0 \leq k \leq N-1)$.

$(\omega^t)^k$ $(\omega^k)^t$ $(\omega^k)^{N-1}$ $\cos\left(\frac{2\pi kt}{N}\right) + i \sin\left(\frac{2\pi kt}{N}\right)$ $(t \in \mathbb{Z}/(N))$

Motivating problem: How can we efficiently express a signal as a linear combination of "sine wave" signals like those?

"pure tone"

Picture behind thinking of a signal f:Z/(N) -> C as a signal of period N:



Repetition every N time units is modelled by the fact that N=0 in Z/(N).

The Discrete Fourier Transform (DFT)

Definition

Fix $N \in \mathbf{N}$, let $\omega = e^{2\pi i/N}$ be the natural primitive Nth root of unity in $\mathbf{C}$, and let $f : \mathbf{Z}/(N) \to \mathbf{C}$ be a signal. We define the DFT of f to be $\hat{f} : \mathbf{Z}/(N) \to \mathbf{C}$ given by

$$\hat{f}(k) = \frac{1}{N} \sum_{n=0}^{N-1} f(n)\omega^{-nk}.$$

Think of $\hat{f}(k)$ as the **spectrum** of f, because (roughly) $\hat{f}(k)$ measures the strength of the part of f that has "frequency k".

Writing out the DFT

Writing out the definition of $\hat{f}(k)$ for $k = 0, 1, 2, 3$, we get

$$\hat{f}(0) = \frac{1}{N}(f(0) + f(1) + f(2) + \cdots + f(N-1)),$$

$$\hat{f}(1) = \frac{1}{N}(f(0) + \omega^{-1} f(1) + \omega^{-2} f(2) + \cdots + \omega^{-(N-1)} f(N-1)),$$

$$\hat{f}(2) = \frac{1}{N}(f(0) + \omega^{-2} f(1) + \omega^{-2(2)} f(2) + \cdots + \omega^{-2(N-1)} f(N-1)),$$

$$\hat{f}(3) = \frac{1}{N}(f(0) + \omega^{-3} f(1) + \omega^{-3(2)} f(2) + \cdots + \omega^{-3(N-1)} f(N-1)).$$

(See PS10.)

The inverse DFT

Definition

Let $\hat{f} : \mathbf{Z}/(N) \to \mathbf{C}$ be a spectrum function. The **inverse DFT** of $\hat{f}$ is defined to be

$$f(n) = \sum_{k=0}^{N-1} \hat{f}(k)\omega^{kn}.$$

We'll show that:

(DFT with a sign change.)

Point of the inverse DFT is:

"pure tone" signal $(\omega^k)^n$

$e_k(n) =$

Theorem (Inversion Theorem)

Fix $N \in \mathbf{N}$, $\omega = \omega_N$, $f : \mathbf{Z}/(N) \to \mathbf{C}$ be a signal. If $\hat{f}$ is the DFT of f, then

$$f(n) = \sum_{k=0}^{N-1} \hat{f}(k)\omega^{kn}.$$

$f \xrightarrow{DFT} \hat{f} \xrightarrow{IDFT} f$

I.e., inverse DFT inverts the DFT.

Proof of Inversion Theorem

Uses (see PS10):

Lemma (Orthogonality Lemma)

Fix $N \in \mathbf{N}$ and let $\omega = \omega_N$. For $t \in \mathbf{Z}/(N)$, we have:

$$\sum_{k=0}^{N-1} \omega^{kt} = \begin{cases} N & \text{if } t = 0 \pmod N, \\ 0 & \text{otherwise.} \end{cases}$$

PS10

Tells us that "pure tones" are at "right angles" in signal space.

Proof of Inversion:

DFT

Want this $= f(x)$

$$\sum_{k=0}^{N-1} \hat{f}(k)\omega^{kx} = \frac{1}{N} \sum_{k=0}^{N-1} \left(\sum_{n=0}^{N-1} f(n)\omega^{-nk} \right) \omega^{kx}$$

ω^{kx-nk}

$$= \frac{1}{N} \sum_{n=0}^{N-1} f(n) \sum_{k=0}^{N-1} \omega^{k(x-n)}$$

Orthogonality: if x=n, this is =N; else =0.

$$= \frac{1}{N} f(x) N = f(x)$$

The killer app: Fast DFT gives fast multiplication

Recall that multiplication of two polynomials of degree N takes time $O(N^2)$.
Thanks to convolution (see Sec. 9.4), we have the following fact:

> **FACT:** *If we can compute the DFT of a period N signal in (for example) time $O(N \log N)$, we can multiply two polynomials in time $O(N \log N)$.*

Can actually adapt algorithm to N-digit numbers, i.e., a fast DFT let us perform ordinary multiplication (of zillion-digit numbers) way way faster. So we ask:

Motivating question: How do we compute the DFT in $O(N \log N)$ time?