Math 127, Wed Apr 28

- Use a laptop or desktop with a large screen so you can read these words clearly.
- In general, please turn off your camera and mute yourself.
- Exception: When we do groupwork, please turn both your camera and mic on. (Groupwork will not be recorded.)
- Please always have the chat window open to ask questions.
- Reading for today: 9.3–9.5. Reading for next Wed: 10.1–10.3.
- PS09 due tonight.
- Exam 3, Mon May 03. PS7-9
- Exam review Fri Apr 30, 10am–noon.

127 10-11

128B 11-noon

Will there be in-person classes in the fall?

Short answer: Yes.

Longer answer: Rooms can only run at 75% capacity, with 30 min breaks between classes instead of 15 min breaks.

Problem: If rooms run at 3/4 capacity, then we need to run 4/3 as many sections. So that means that many smaller classes will run online, or at least hybrid.

"Hybrid" often = class online, but exams in person.

Questions?

Recap/foreshadowing: What you really need to know about ω

$e^{i\alpha} e^{i\beta} = e^{i(\alpha+\beta)}$

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. Fact:

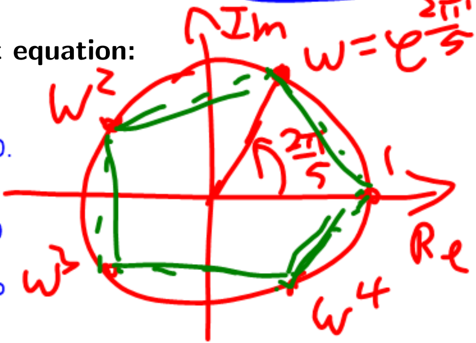
$$\boxed{1 + \omega + \cdots + \omega^{N-1} = 0.}$$

$\omega^N = 1$

Why the last equation:

Nth roots of unity form a regular N-gon with center 0.

Average of these numbers must be 0 (by symmetry), so their sum must also be 0.



$N = 5$

Signals

Definition

Fix $N \in \mathbf{N}$. We define a **signal** to be 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 in vector form as $\begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}$.

$e_2(n) = \omega^{2n}$

Example: Let $\omega = e^{2\pi i/N}$ be the natural primitive Nth root of unity in $\mathbf{C}$. We define the **basic trigonometric signal** $e_k : \mathbf{Z}/(N) \to \mathbf{C}$ by $e_k(n) = \omega^{kn}$. We can also represent e_k in vector form as $\begin{bmatrix} 1 \\ \omega^k \\ \vdots \\ \omega^{(N-1)k} \end{bmatrix}$.

$e_0(n) = \omega^{0 \cdot n} = 1$

$e_1(n) = \omega^n$

Examples: e_k for $N = 12$, $k = 0, 1, 2, 3, \cancel{4}$

$w^{12} = 1$

$$
\begin{array}{c|c|c|c|c}
 & e_0 & e_1 & e_2 & e_3 \\
\hline
n=0 & 1 & 1 & 1 & 1 \\
1 & 1 & w & w^2 & w^3 \\
2 & 1 & w^2 & w^4 & w^6 \\
3 & 1 & w^3 & w^6 & w^9 \\
4 & 1 & w^4 & w^8 & 1 \\
5 & 1 & w^5 & w^{10} & w^3 \\
6 & 1 & w^6 & 1 & w^6 \\
7 & 1 & w^7 & w^2 & w^9 \\
8 & 1 & w^8 & w^4 & 1 \\
9 & 1 & w^9 & w^6 & w^3 \\
10 & 1 & w^{10} & w^8 & w^6 \\
n=11 & 1 & w^{11} & w^{10} & w^9
\end{array}
$$

e_1 freq 1

e_2 freq 2

e_3 freq 3

e_1

e_2

e_3

Orthogonality Lemma

Fix $N \in \mathbf{N}$ and let $\omega = \omega_N = e^{2\pi i/N}$ be the natural primitive Nth root of unity in $\mathbf{C}$. For $t \in \mathbf{Z}/(N)$, we have:

Sum of the values of the vectors from the previous slide!

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

Proof: See PS10.

In particular, if $t = 1$:, Lemma says:

$$0 = \sum_{k=0}^{N-1} w^k = w^0 + w^1 + w^2 + \cdots + w^{N-1}$$

$$= 1 + w^1 + w^2 + \cdots + w^{N-1}$$

A motivating problem

$$f = \begin{bmatrix} 1 \\ 7 \\ \pi + i \\ \vdots \\ \vdots \end{bmatrix} \stackrel{?}{=} a_0 e_0 + a_1 e_1 + \cdots$$

Motivating Problem

Fix $N \in \mathbf{N}$. How can we express any signal on $\mathbf{Z}/(N)$ as a linear combination of the basic trigonometric signals e_k, $0 \leq k \leq N-1$?

Solving this problem has many applications (e.g., analysis of music/sound production) but we'll concentrate on one: making multiplication faster. (!!)

of v. large integers!!

See: ProTools and other digital music software.

The Discrete Fourier Transrom

form

$\omega^N = 1$

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

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 **Discrete Fourier Transform**, or **DFT**, of f to be the function $\hat{f} : \mathbf{Z}/(N) \to \mathbf{C}$ given by

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

(Think of $\hat{f}(k)$ not as a signal, but as the "spectrum" of f.)

$\hat{f}(k)$ = how much of f is in freq k.

Example: DFT for $N = 4$

$w^4 = 1,\quad 1 + \cdots + w^3 = 0$

$k = 0, 1, 2, 3$

$$\hat{f}(0) = \frac{1}{4}\sum_{n=0}^{3} f(n) w^0 = \frac{1}{4}\left(f(0) + f(1) + f(2) + f(3)\right)$$

$$\hat{f}(1) = \frac{1}{4}\sum_{n=0}^{3} f(n) w^{-n}$$

$$= \frac{1}{4}\left(f(0) + f(1)w^{-1} + f(2)w^{-2} + f(3)w^{-3}\right)$$

$$\hat{f}(2) = \frac{1}{4}\sum_{n=0}^{3} f(n) w^{-2n}$$

PS10: Do this for N = 6, 8, 9, or 12.

$$= \frac{1}{4}\left(f(0) + f(1)w^{-2} + f(2) + f(3)w^{-2}\right)$$

$n=0 \qquad n=1 \qquad n=2$

DFT in matrix form

$$
\begin{bmatrix} \hat{f}(0) \\ \vdots \\ \hat{f}(N-1) \end{bmatrix}
= \frac{1}{N} \begin{bmatrix}
1 & 1 & 1 & \dots & 1 \\
1 & \omega^{-1} & \omega^{-2} & \dots & \omega^{-(N-1)} \\
1 & \omega^{-2} & \omega^{-2(2)} & \dots & \omega^{-2(N-1)} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \dots & \omega^{-(N-1)(N-1)}
\end{bmatrix}
\begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}.
$$

The point: Applying the DFT is matrix-vector multiplication, and therefore, $\cancel{O(n^2)}$. $O(N^2) \xrightarrow{\text{FFT}} O(N \log N)$

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

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

Basically the same as the DFT, but with a sign change and without the $\frac{1}{N}$. However:

Theorem (Inversion Theorem)

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. If $\hat{f}$ is the DFT of f, then

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

We get the original signal back.

Matrix-vector version of inverse ~~DFT~~ Thm

$\hat{T}$

$$\begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix} = \begin{array}{c} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega^1 & \omega^2 & \dots & \omega^{(N-1)} \\ 1 & \omega^2 & \omega^{2(2)} & \dots & \omega^{(N-1)2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{(N-1)} & \omega^{2(N-1)} & \dots & \omega^{(N-1)(N-1)} \end{bmatrix} \\ \begin{array}{ccccc} e_0 & e_1 & e_2 & \quad & e_{N-1} \end{array} \end{array} \begin{bmatrix} \hat{f}(0) \\ \vdots \\ \hat{f}(N-1) \end{bmatrix}$$

I.e.: If $\hat{T}$ has kth column e_k, then $\hat{T} \begin{bmatrix} \hat{f}(0) \\ \vdots \\ \hat{f}(N-1) \end{bmatrix} = \begin{bmatrix} f(0) \\ \vdots \\ f(N-1) \end{bmatrix}$.

Since this is the linear combination of the columns of $\hat{T}$ with coefficients taken from $\begin{bmatrix} \hat{f}(0) \\ \vdots \\ \hat{f}(N-1) \end{bmatrix}$, we see that the $\hat{f}$ are the coeffs that express f as a lin comb of the basic trig signals e_k.

Example: Inverse DFT for $N = 4$

$$\boxed{w^4 = 1}$$

Point: If we can compute DFT quickly, we can compute inverse DFT quickly.

$n = 1$

$$\sum_{k=0}^{3} \hat{f}(k) w^k$$

$$= \hat{f}(0) + \hat{f}(1) w^1 + \hat{f}(2) w^2 + \hat{f}(3) w^3$$

$$= \cdots = f(1)$$

$n = 2$

$$\sum_{k=0}^{3} \hat{f}(k) w^{2k} = \hat{f}(0) + \hat{f}(1) w^2 + \hat{f}(2) + \hat{f}(3) w^2$$

Convolution (one application of the DFT)

We now explain why, if you were able to ~~complete~~ compute the DFT quickly, it would lead to a fast multiplication algorithm, or something pretty close to it. First:

Definition

Fix $N>0$.

Let $f, g : \mathbf{Z}/(N) \to \mathbf{C}$ be signals. We define the **convolution** of f and g to be the signal $f * g : \mathbf{Z}/(N) \to \mathbf{C}$ defined by

$$(f * g)(n) = \frac{1}{N} \sum_{t=0}^{N-1} f(n-t)g(t).$$

Convolution of signals is polynomial multiplication

so $O(N^2)$

Theorem

Let $f, g : \mathbf{Z}/(N) \to \mathbf{C}$ be signals. Then in the ring $\mathbf{C}[x]/(x^N - 1)$, we have that

$$\left(\frac{1}{N}\sum_{k=0}^{N-1} f(k)x^k\right)\left(\frac{1}{N}\sum_{m=0}^{N-1} g(m)x^m\right) = \frac{1}{N}\sum_{n=0}^{N-1}(f*g)(n)x^n.$$

The details aren't crucial — the point is, if you want to multiply two complex polynomials mod $(x^N - 1)$ it's enough to compute the convolution of the corresponding signals.

Real motivation: If N is several times larger than the degrees of f and g, multiplication mod $x^N - 1$ is the same as multiplication of f and g, which is pretty close to multiplying two integers.

DFT(convolution) = pointwise product of DFTs

Theorem

Let $f, g : \mathbf{Z}/(N) \to \mathbf{C}$ be signals. We have that

$$\widehat{(f * g)}(k) = \hat{f}(k)\hat{g}(k).$$

So if we know $\hat{f}(k)$ and $\hat{g}(k)$, we just do the above multiplication N times to find $\widehat{(f * g)}(k)$ for $0 \leq k \leq N - 1$. This kind of "pointwise product" is an $O(N)$ procedure.

An algorithm for fast polynomial multiplication

Motivating Problem

Compute the product of two polynomials in $\mathbf{C}[x]/(x^N - 1)$ whose coefficients are given by $f(n)$ and $g(n)$. In other words, given two signals $f(n)$ and $g(n)$, compute the convolution $(f * g)(n)$.

Note that polynomial multiplication is usually $O(N^2)$.

First attempted algorithm. Suppose we have two signals $f, g : \mathbf{Z}/(N) \to \mathbf{C}$. representing coeffs of a polynomial

$O(N^2)$ 1. Compute the DFTs $\hat{f}(k)$ and $\hat{g}(k)$.

$O(N)$ 2. For all $k \in \mathbf{Z}/(N)$, let $\hat{h}(k) = \hat{f}(k)\hat{g}(k)$.

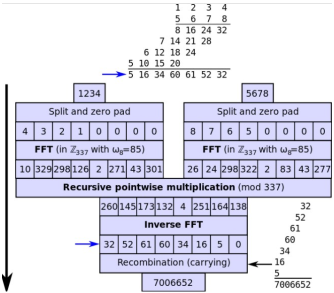
$O(N^2)$ 3. Compute the inverse DFT $h(n)$ of $\hat{h}(k)$. ← Thm: $h = f * g$

Step (2) is $O(N)$, but if we compute the DFT through standard matrix multiplication, the other two steps are still $O(N^2)$.

The punchline

FFT

By the end of the course, we'll see an algorithm, called the **Fast Fourier Transform**, that computes the DFT in $O(N \log N)$ time. This gives an algorithm for multiplying polynomials of degree N that is $2 * O(N \log N) + O(N) = O(N \log N)$. In fact, Schönhage and Strassen turned this into an algorithm for multiplying N-digit integers that is $O(N \log N \log(\log N))$:



New Alg

1. $O(N \log N)$

2. $O(N)$

3. $O(N \log N)$

But to understand the FFT, we first need to understand **groups**.