The Discrete Fourier Transform

MATH 420 & CYBR 304

Spring 2024

"The profound study of nature is the most fertile source of mathematical discovery." — Joseph Fourier

A trigonometric basis

Let $n \in \mathbf{Z}_{\geq 0}$ and let a_0 through a_n and b_1 though b_n be real numbers. Define a function F by

$$F(x) = a_0 + \sum_{k=1}^{n} a_k \cos(kx) + a_k \sin(kx).$$

Expressing cos and sin in exponential form, the formula is

$$\begin{aligned} F(x) &= a_0 + \sum_{k=1}^{n} a_k \frac{\mathrm{cis}(kx) + \overline{\mathrm{cis}(kx)}}{2} + b_k \frac{\mathrm{cis}(kx) - \overline{\mathrm{cis}(kx)}}{2\mathrm{i}}, \\ &= a_0 + \sum_{k=1}^{n} \frac{1}{2}(a_k - \mathrm{i} b_k)\mathrm{cis}(kx) + \frac{1}{2}(a_k + \mathrm{i} b_k)\overline{\mathrm{cis}(kx)}. \end{aligned}$$

An exponential basis

And

$$F(x) = a_0 + \sum_{k=1}^{n} \frac{1}{2}(a_k - \mathrm{i}b_k)\mathrm{cis}(kx) + \frac{1}{2}(a_k + \mathrm{i}b_k)\overline{\mathrm{cis}(kx)}$$
$$= a_0 + \mathrm{Re}\left(\sum_{k=1}^{n}(a_k - \mathrm{i}b_k)\mathrm{cis}(kx)\right)$$

Defining complex numbers $c_0, c_1, \ldots, c_n$ as

$$c_k = \begin{cases} a_0 & k = 0 \\ a_k - \mathrm{i}b_k & k \neq 0 \end{cases}$$

we have

$$F(x) = \mathrm{Re}\left(\sum_{k=0}^{n} c_k \,\mathrm{cis}(kx)\right).$$

Unreal functions

If we drop the condition that F is real-valued and use the fact that $\overline{\mathrm{cis}(kx)} = \mathrm{cis}(-kx)$ we have

$$
\begin{aligned}
F(x) &= a_0 + \sum_{k=1}^{n} \frac{1}{2}(a_k - \mathrm{i}b_k)\mathrm{cis}(kx) + \frac{1}{2}(a_k + \mathrm{i}b_k)\mathrm{cis}(-kx), \\
&= \sum_{k=-n}^{n} c_k \mathrm{cis}(kx).
\end{aligned}
$$

where

$$
c_k = \begin{cases} \frac{1}{2}(a_k + \mathrm{i}b_k) & k < 0 \\ a_0 & k = 0 \\ \frac{1}{2}(a_k - \mathrm{i}b_k) & k > 0 \end{cases}
$$

Unreal and orthogonal

For any $n \in \mathbf{Z}_{\geq 0}$, the set of functions

$$\{x \mapsto \operatorname{cis}(kx) \mid k \in -n \dots n\}$$

is orthogonal with respect to the inner product

$$\langle f, g \rangle = \int_0^{2\pi} \overline{f(x)} g(x) \, \mathrm{d}x.$$

In particular

$$\int_0^{2\pi} \overline{\operatorname{cis}(kx)} \operatorname{cis}(\ell x) \mathrm{d}x = \begin{cases} 0 & k \neq \ell \\ 2\pi & k = \ell \end{cases} = 2\pi \delta_{k,\ell}$$

Let $n \in \mathbf{Z}_{\geq 0}$ and let $c_{-n}, \ldots, c_n \in \mathbf{C}$. Define a function F whose formula is

$$F(x) = \sum_{k=-n}^{n} c_k \mathrm{e}^{\mathrm{i}kx} \tag{1}$$

For a given function F, we would like to find a way to find the complex numbers $c_{-n}, \ldots, c_n$.

Step #1 Multiply Eq. 1 by $\mathrm{e}^{-\mathrm{i}\ell x}$. This gives

$$\mathrm{e}^{-\mathrm{i}\ell x} F(x) = \sum_{k=-n}^{n} c_k \mathrm{e}^{-\mathrm{i}\ell x} \mathrm{e}^{\mathrm{i}kx} \tag{2}$$

Using a rule of exponents, we have

$$\mathrm{e}^{-\mathrm{i}\ell x} F(x) = \sum_{k=-n}^{n} c_k \mathrm{e}^{-\mathrm{i}(k-\ell)x}. \tag{3}$$

Step #2 Integrate with respect to x over the interval $[0, 2\pi]$

$$\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x)\,\mathrm{d}x = \int_0^{2\pi} \sum_{k=-n}^{n} c_k \mathrm{e}^{-\mathrm{i}(k-\ell)x}\,\mathrm{d}x. \tag{4}$$

Swap the integration and the finite sum:

$$\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x)\,\mathrm{d}x = \sum_{k=-n}^{n} c_k \int_0^{2\pi} \mathrm{e}^{-\mathrm{i}(k-\ell)x}\,\mathrm{d}x. \tag{5}$$

Step #3 Integrate the orthogonal functions:

$$\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x)\,\mathrm{d}x = \sum_{k=-n}^{n} c_k 2\pi \delta_{k,\ell}. \tag{6}$$

Simplify the sum

$$\int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x)\,\mathrm{d}x = 2\pi c_\ell. \tag{7}$$

We've shown that if

$$F(x) = \sum_{k=-n}^{n} c_k \mathrm{e}^{\mathrm{i}kx} \tag{8}$$

Then for all $\ell \in -n \ldots n$, we have

$$c_\ell = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x)\,\mathrm{d}x \tag{9}$$

- ☛ Given the function F, the complex numbers c_{-n} through c_n are uniquely determined.

I know what you are thinking

Unless F is fairly simple, we have no chance at finding a formula for the numbers c_{-n} through c_n.

Have no fear: we have a tool for that. Let's use the right-point rule integration to find approximate values for the Fourier coefficients. Using n equal length subintervals of $[0, 2\pi]$, we have

$$\begin{aligned} c_\ell &\approx \frac{1}{2\pi}\frac{2\pi}{n}\sum_{k=0}^{n-1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{n}\ell k} F(x_k) \\ &= \frac{1}{n}\sum_{k=0}^{n-1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{n}\ell k} F\left(\frac{2\pi}{n}k\right) \end{aligned}$$

Caveat

For large values of the integer ℓ, the integrand of

$$c_\ell = \frac{1}{2\pi} \int_0^{2\pi} \mathrm{e}^{-\mathrm{i}\ell x} F(x) \, \mathrm{d}x$$

becomes more and more "wiggly." So we should expect that our right point rule quadrature rule will give less and less accurate results for larger and larger ℓ.

DFT defined

Definition Let $a_0, a_1, \dots a_{n-1}$ be numbers (either real or complex). For every $\ell \in 0 \dots n-1$, define

$$\widehat{a}_\ell = \sum_{k=0}^{n-1} \mathrm{e}^{-\mathrm{i}\frac{2\pi}{n}\ell k} a_k$$

The list of numbers $\widehat{a}_0, \widehat{a}_1, \dots, \widehat{a}_{n-1}$ is the *discrete Fourier transform* (DFT) of $a_0, a_1, \dots, a_{n-1}$.

Doubly indexed things are matrices

For any $n \in \mathbf{Z}_{>0}$, define $\mathcal{F}_{\ell,k} = \mathrm{e}^{-\mathrm{i}\frac{2\pi}{n}\ell k}$. We have

$$\widehat{a}_\ell = \sum_{k=0}^{n-1} \mathcal{F}_{\ell,k} a_k$$

we see that the DFT is matrix multiplication. Arranging $\widehat{a}_0, \ldots \widehat{a}_{n-1}$ and $a_0, \ldots \widehat{a}_{n-1}$ as column vectors $\mathbf{a}$ and $\widehat{\mathbf{a}}$, we have

$$\widehat{\mathbf{a}} = \mathcal{F}\,\mathbf{a}$$

Magic properties

- When the size of a matrix $\mathcal{F}$ is $n \times n$, ordinarily the effort required to do the matrix product $\mathcal{F}\,\mathbf{a}$ is proportional to n^2.
- But the matrix $\mathcal{F}$ has some special (almost magic) properties that makes the effort only proportional to $n \log_{10} n$.
- The algorithm that uses these properties to do the multiplication superfast is the *Fast Fourier Transform* (FFT).
- The FFT isn't really a transform, but an algorithm that computes a transform.
- When n is large, $n^2 \gg n \log_{10}(n)$.