Fourier Transform

L05

Goal: To introduce some of the basic methods for the Fourier transform.

Fourier Transform (Continuous Domain)

Let $f: \mathbb{R} \to \mathbb{R}$ (i.e. $f(x)$)

Its Fourier transform (FT) is defined as

$$F(\omega) = \mathcal{F}\{f(x)\}(\omega) = \int_{-\infty}^{\infty} f(x)\, e^{-2\pi i \omega x}\, dx$$

$F(\omega)$ is a frequency decomposition... a different way of representing the same signal f.

The FT is invertible

$$f(x) = \mathcal{F}^{-1}\{F(\omega)\}(x) = \int_{-\infty}^{\infty} F(\omega)\, e^{2\pi i \omega x}\, d\omega$$

Discrete Fourier Transform (DFT)

When both $f(x)$ and $F(\omega)$ are sampled,

i.e. $f_n,\ n = 0, \dots, N-1 \Rightarrow [f_0 \ f_1 \ f_2 \ \dots \ f_{N-1}]$

$F_k,\ k = 0, \dots, N-1 \Rightarrow [F_0 \ F_1 \ F_2 \ \dots \ F_{N-1}]$

then we have the Discrete Fourier Transform (DFT) and its inverse:

$$F_k = \sum_{n=0}^{N-1} f_n\, e^{\frac{-2\pi i n k}{N}} \qquad f_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k\, e^{\frac{2\pi i n k}{N}}$$

$k = 0, \dots, N-1 \qquad n = 0, \dots, N-1$

Fast Fourier Transform (FFT)

Computing all N Fourier coefs. directly using the formula for F_k above would take $O(N^2)$ flops.

A divide-and-conquer method turns out to be faster. It's called the Fast Fourier Transform.

Briefly, it decomposes the length-N DFT into two $\frac{N}{2}$-length DFTs. This process recursively decomposes the DFT until it arrives at arrays of length 1 ... those are easy. The whole process takes $O(N \log N)$ flops for a 1D array of length N.

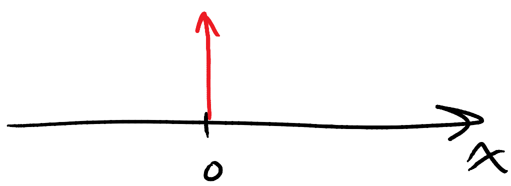
(what about for 2D or 3D?)

Given f_n, $n = 0, \dots, N-1$, we recombine the elements to get g_n and h_n, $n = 0, \dots, \frac{N}{2}-1$.

$$\begin{tikzcd}[column sep=small]
 & & \mathrm{DFT}(gg, \frac{N}{4}) \\
 & \mathrm{DFT}(g, \frac{N}{2}) \arrow[ur] \arrow[dr] & \\
 & & \mathrm{DFT}(hg, \frac{N}{4}) \quad \cdots \\
\mathrm{DFT}(f, N) \arrow[uur] \arrow[ddr] & & \\
 & & \mathrm{DFT}(gh, \frac{N}{4}) \\
 & \mathrm{DFT}(h, \frac{N}{2}) \arrow[ur] \arrow[dr] & \\
 & & \mathrm{DFT}(hh, \frac{N}{4})
\end{tikzcd}$$

Def$^{\underline{n}}$: Dirac delta function

$$\delta(x) = \begin{cases} \infty & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$$



s.t. $\displaystyle\int_{-\infty}^{\infty} \delta(x)\,dx = 1$

and $\displaystyle\int_{-\infty}^{\infty} \delta(x-c)\, f(x)\,dx = f(c)$

Theorem: $$\int_{-\infty}^{\infty} e^{2\pi i \omega x}\,dx = \delta(\omega)$$

Pf: Not in THIS course.

We can use the Dirac delta function to prove that $\mathcal{F}$ and $\mathcal{F}^{-1}$ are inverses.

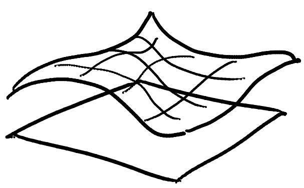
Pf: $\mathcal{F}^{-1}\left\{\mathcal{F}\{f(x)\}(\omega)\right\}(s)$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x)e^{-2\pi i \omega x}\,dx\, e^{2\pi i \omega s}\,d\omega$$

$$= \int_{-\infty}^{\infty} f(x)\left[\int_{-\infty}^{\infty} e^{2\pi i \omega (s-x)}\,d\omega\right]dx$$

$$= \int_{-\infty}^{\infty} f(x)\,\delta(s-x)\,dx = f(s)$$

2D Fourier Transform



Consider the function $f: \mathbb{R}^2 \to \mathbb{R}$

The 2D FT is defined as

$$F(\omega,\lambda) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\,e^{-2\pi i \underbrace{(\omega x+\lambda y)}_{(\omega,\lambda)\cdot(x,y)}}\,dx\,dy$$

Notice that the FT is separable:

$$F(\omega,\lambda) = \int\int f(x,y)\,e^{-2\pi i \omega x}\,dx\;e^{-2\pi i \lambda y}\,dy$$

$$= \underbrace{\int \underbrace{\left[\int f(x,y)\, e^{-2\pi i \omega x}\, dx\right]}_{\text{Apply FT along } x\text{-dim}} e^{-2\pi i \lambda y}\, dy}_{\text{Apply FT along } y\text{-dim}}$$

Thus, an N-D FT can be done using 1D FTs along each of the N dimensions.

TASK: Check out the associated Matlab script. There are some lines for you to fill in, and a number of important concepts to learn.