The Discrete Fourier Transform

DS-GA 1013 / MATH-GA 2824 Mathematical Tools for Data Science

Carlos Fernandez-Granda

Prerequisites

Calculus (complex numbers)

Linear algebra (orthogonality, basis, projections)

Fourier series

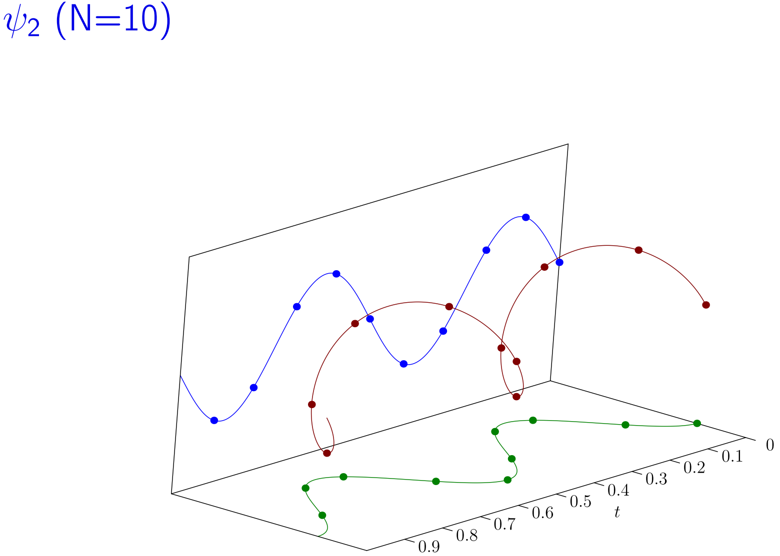
The sampling theorem

Discrete complex sinusoids

The discrete complex sinusoid $\psi_k \in \mathbb{C}^N$ with frequency k is

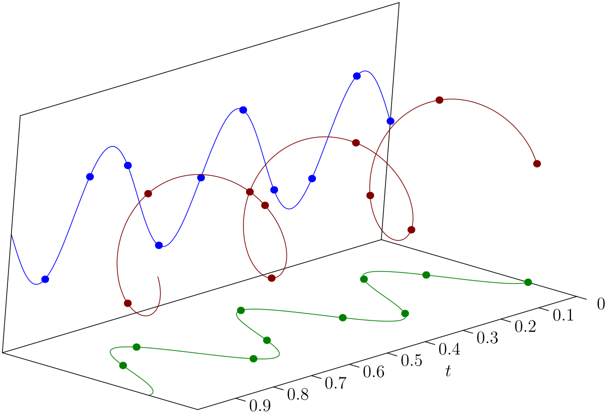
$$\psi_k [j] := \exp \left(\frac{i 2\pi k j}{N} \right), \qquad 0 \leq j, k \leq N - 1$$

Discrete complex sinusoids scaled by $1/\sqrt{N}$: orthonormal basis of $\mathbb{C}^N$



ψ_2 (N=10)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
t

ψ_3 (N=10)



Discrete Fourier transform

The discrete Fourier transform (DFT) of $x \in \mathbb{C}^N$ is

$$\begin{aligned}\hat{x} &:= \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \exp\left(-\frac{i2\pi}{N}\right) & \exp\left(-\frac{i2\pi 2}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\right) \\ 1 & \exp\left(-\frac{i2\pi 2}{N}\right) & \exp\left(-\frac{i2\pi 4}{N}\right) & \cdots & \exp\left(-\frac{i2\pi 2(N-1)}{N}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \exp\left(-\frac{i2\pi(N-1)}{N}\right) & \exp\left(-\frac{i2\pi 2(N-1)}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)^2}{N}\right) \end{bmatrix} x \\ &= F_{[N]} x\end{aligned}$$

$$\hat{x}[k] = \langle x, \psi_k \rangle, \qquad 0 \leq k \leq N-1$$

Inverse discrete Fourier transform

The inverse DFT of a vector $\hat{y} \in \mathbb{C}^N$ equals

$$y = \frac{1}{N} F_{[N]}^* \hat{y}$$

It inverts the DFT

$\frac{1}{\sqrt{N}} F_{[N]}$ has orthonormal columns, so

$$\frac{1}{N} F_{[N]}^* F_{[N]} = I$$

Interpretation in terms of bandlimited signals

If $x \in \mathbb{C}^N$ contains samples of a bandlimited signal such that $2k_c + 1 \leq N$ the DFT is proportional to the Fourier series coefficients of the function

Sampling a bandlimited signal on a uniform grid

Using Fourier series

$$x\left(\frac{jT}{N}\right) = \frac{1}{T}\sum_{k=-k_c}^{k_c} \hat{x}_k \exp\left(\frac{i2\pi k jT}{NT}\right)$$

$$= \frac{1}{T}\sum_{k=-k_c}^{k_c} \hat{x}_k \exp\left(\frac{i2\pi kj}{N}\right)$$

$$= \frac{1}{T}\sum_{k=-k_c}^{k_c} \hat{x}_k \psi_k[j]$$

Vector of samples equals

$$x_{[N]} = \frac{1}{T}\sum_{k=-k_c}^{k_c} \hat{x}_k \psi_k$$

Sampling a bandlimited signal on a uniform grid

How do we recover the Fourier coefficients assuming $N = 2k_c + 1$?

$$x_{[N]} = \frac{1}{T} \sum_{k=-k_c}^{k_c} \hat{x}_k \psi_k$$

$$\begin{aligned}
\frac{T}{N} \langle x_{[N]}, \psi_k \rangle &= \frac{T}{N} \left\langle \frac{1}{T} \sum_{m=-k_c}^{k_c} \hat{x}_m \psi_m, \psi_k \right\rangle \\
&= \sum_{m=-k_c}^{k_c} \hat{x}_m \left\langle \frac{1}{\sqrt{N}} \psi_m, \frac{1}{\sqrt{N}} \psi_k \right\rangle \\
&= \hat{x}_k
\end{aligned}$$

Complexity of computing the DFT

Complexity of multiplying $N \times N$ matrix with N-dimensional vector is N^2

Very slow!

We can exploit the structure of the matrix to do much better

Fast Fourier transform

The most important numerical algorithm of our lifetime (G. Strang)

Main insight:

$N \times N$ DFT matrix can be expressed in terms of $\frac{N}{2} \times \frac{N}{2}$ DFT matrices

Separation in even/odd columns and top/bottom rows

For simplicity, assume N is even

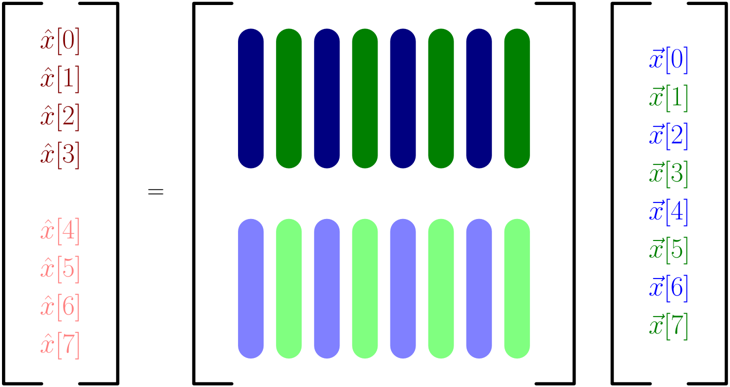
$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \exp\left(-\frac{i2\pi}{N}\right) & \exp\left(-\frac{i2\pi 2}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \exp\left(-\frac{i2\pi}{N}\frac{N-1}{2}\right) & \exp\left(-\frac{i2\pi 2}{N}\frac{N-1}{2}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\frac{N-1}{2}\right) \\ 1 & \exp\left(-\frac{i2\pi}{N}\frac{N}{2}\right) & \exp\left(-\frac{i2\pi 2}{N}\frac{N}{2}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\frac{N}{2}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \exp\left(-\frac{i2\pi(N-1)}{N}\right) & \exp\left(-\frac{i2\pi 2(N-1)}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)^2}{N}\right) \end{bmatrix}$$

Separation in even/odd columns and top/bottom rows

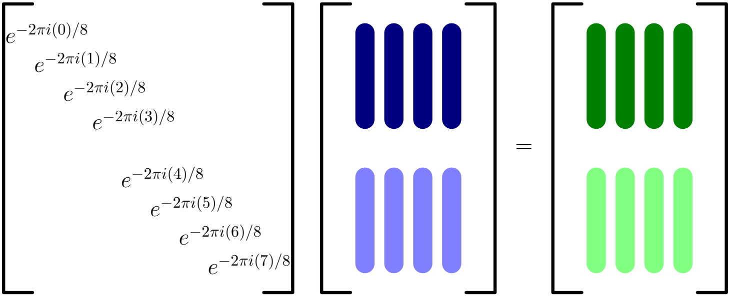
For simplicity, assume N is even

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \exp\left(-\frac{i2\pi}{N}\right) & \exp\left(-\frac{i2\pi 2}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \exp\left(-\frac{i2\pi}{N}\frac{N-1}{2}\right) & \exp\left(-\frac{i2\pi 2}{N}\frac{N-1}{2}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\frac{N-1}{2}\right) \\ 1 & \exp\left(-\frac{i2\pi}{N}\frac{N}{2}\right) & \exp\left(-\frac{i2\pi 2}{N}\frac{N}{2}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)}{N}\frac{N}{2}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & \exp\left(-\frac{i2\pi(N-1)}{N}\right) & \exp\left(-\frac{i2\pi 2(N-1)}{N}\right) & \cdots & \exp\left(-\frac{i2\pi(N-1)^2}{N}\right) \end{bmatrix}$$

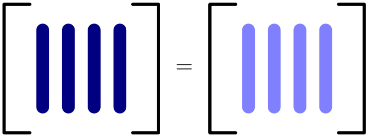
Separation in even/odd columns and top/bottom rows



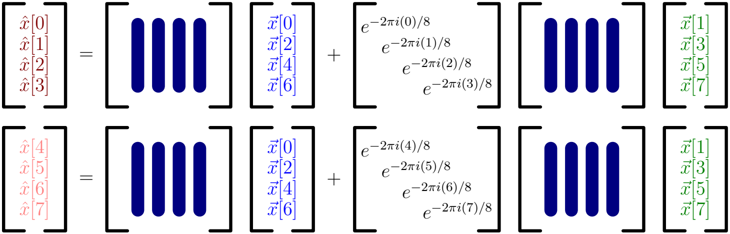
Even columns can be scaled to yield odd columns



Top even submatrix and bottom even submatrix are both an $\frac{N}{2} \times \frac{N}{2}$ DFT matrix



FFT identity



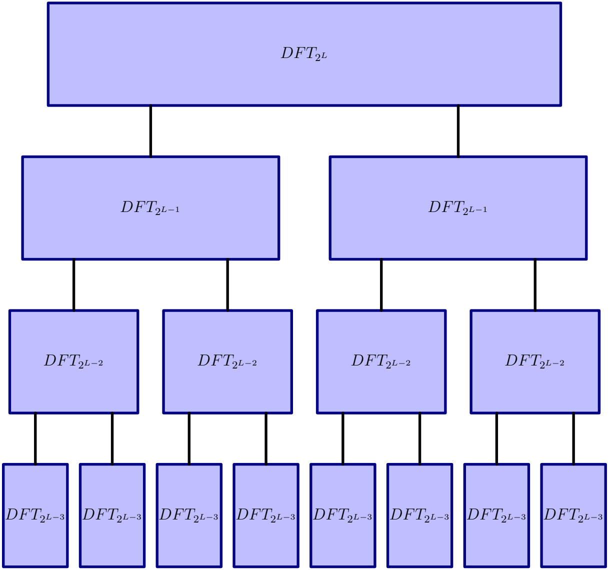
Cooley-Tukey Fast Fourier transform

1. Compute $F_{[N/2]} x_{\text{even}}$.
2. Compute $F_{[N/2]} x_{\text{odd}}$.
3. For $k = 0, 1, \ldots, N/2 - 1$ set

$$F_{[N]} x\,[k] := F_{[N/2]} x_{\text{even}}\,[k] + \exp\left(-\frac{i2\pi k}{N}\right) F_{[N/2]} x_{\text{odd}}\,[k]$$

$$F_{[N]} x\,[k + N/2] := F_{[N/2]} x_{\text{even}}\,[k] - \exp\left(-\frac{i2\pi k}{N}\right) F_{[N/2]} x_{\text{odd}}\,[k]$$

Complexity



DFT_{2^L}
$DFT_{2^{L-1}}$
$DFT_{2^{L-1}}$
$DFT_{2^{L-2}}$
$DFT_{2^{L-2}}$
$DFT_{2^{L-2}}$
$DFT_{2^{L-2}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$
$DFT_{2^{L-3}}$

Complexity

Assume $N = 2^L$

$L = \log_2 N$ levels

At level $m \in \{1, \ldots, L\}$ there are 2^m nodes

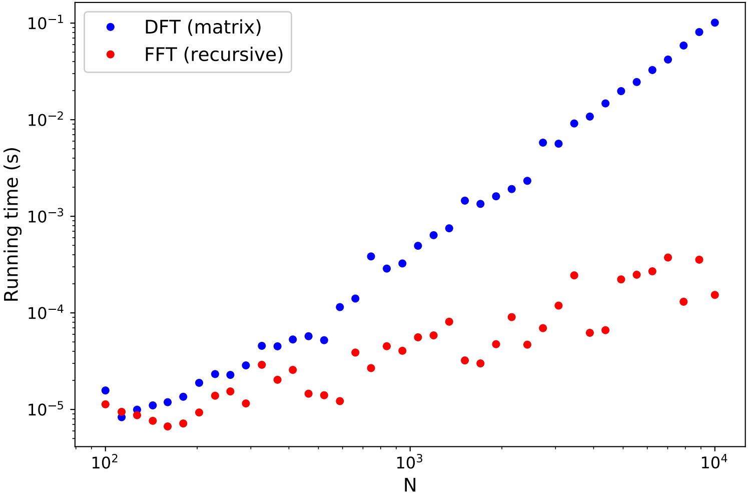
At each node, scale a vector of dim 2^{L-m} and add to another vector

Complexity at each node: 2^{L-m}

Complexity at each level: $2^{L-m}2^m = 2^L = N$

Complexity is $O(N \log N)$!

In practice



DFT (matrix)
FFT (recursive)
Running time (s)
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{2}
10^{3}
10^{4}
N

What have we learned

Definition of discrete Fourier transform

How to compute it efficiently using the fast Fourier transform