# Discrete Fourier Transform (DFT)

$O(n^2)$ computation (vector times matrix).

# The Fast Fourier Transform (FFT)

is a method of computing the DFT,
but much faster : $O(n \log(n))$.

One of the most important numerical algorithms
ever created!

Used every time you :
- Make a cell phone call
- take a picture and save as jpeg.
- streaming video
- ... and so much more!

→ Cooley & Tukey 1965 (IBM, Princeton)

→ Actually discovered by Gauss in 1805!

Example

10 seconds of audio:

Audio is sampled at 44.1 kHz

= 44100 samples per second

10 seconds audio = $4.41 \times 10^5 = n$.

| DFT multiplications | FFT mults | Speed up |
| --- | --- | --- |
| $\sigma(n^2)$ | $\sigma(n \log(n))$ | |
| 2 billion | 1/4 million. | 8000× |

- cell phones and ipods wouldn't work without FFT...

# Fast Fourier Transform (FFT)

DFT may be implemented much more efficiently if number of data points is a power of 2:

$$\underbrace{x_0, x_1, x_2, \cdots, x_{1023}}_{1024 \text{ entries} = 2^{10}}.$$

$F_{1024}$ is matrix taking $\{x\} \to \{\hat{x}\}$.

$$\underline{\hat{x}} = \underline{\underline{F}}_{1024}\, \underline{x}$$

$$\underline{\hat{x}} = \begin{bmatrix} I & -D \\ I & -D \end{bmatrix} \begin{bmatrix} \underline{\underline{F}}_{512} & 0 \\ 0 & \underline{\underline{F}}_{512} \end{bmatrix} \begin{bmatrix} \underline{x}_{even} \\ \underline{x}_{odd} \end{bmatrix}$$

where $D = \begin{bmatrix} 1 & & & 0 \\ & w & & \\ & & w^2 & \\ 0 & & & \ddots \\ & & & & w^{511} \end{bmatrix}$

and $\underline{\underline{F}}_{512}$ is broken into $\underline{\underline{F}}_{256} \to \underline{\underline{F}}_{128} \to \underline{\underline{F}}_{64} \to \underline{\underline{F}}_{32} \cdots$.

Even if number of data points is not $= 2^p$, pad with zeros until it is.

[Gilbert Strang has a great lecture on FFT (MIT open course ware #31).]

$\mathcal{O}(n \log(n))$ is better than $\mathcal{O}(n^2)$