The Fourier Transform

CS 355: Introduction to Graphics and Image Processing

First, a bit about
sines and cosines…

Sinusoids

- Useful to think of sinusoids in terms of three properties:
  - their *frequency* (how often they repeat)
  - their *amplitude* (height of the peaks)
  - their *phase* (shifting left or right)
- A sine wave is just a cosine wave with phase π/2

Amplitude

Frequency

$$A \ \cos(2\pi u t + p)$$

Phase (shift)

And a bit about transforms…

Functions as Vectors

- Inner (dot) product between two vectors is the summation of the point-wise product

$$\mathbf{u} \cdot \mathbf{v} = \sum_j \mathbf{u}[j] \; \mathbf{v}[j]$$

- Can't we do the same thing with functions?
- For continuous functions, the summation just becomes an integral

$$f(t) \cdot g(t) = \int_{-\infty}^{\infty} f(t) \; g(t) \; dt$$

- Functions satisfy all of the mathematical requirements for "vectors"
- Can we transform *functions*?

Sinusoidal Basis Functions

- One set of orthonormal basis functions is the set of sines and cosines of different frequencies
- Let's use these as the basis functions for a transform

$$c(u) = \int_{-\infty}^{\infty} f(t) \ \cos(2\pi u t) \ dt$$

$$s(u) = \int_{-\infty}^{\infty} f(t) \ \sin(2\pi u t) \ dt$$

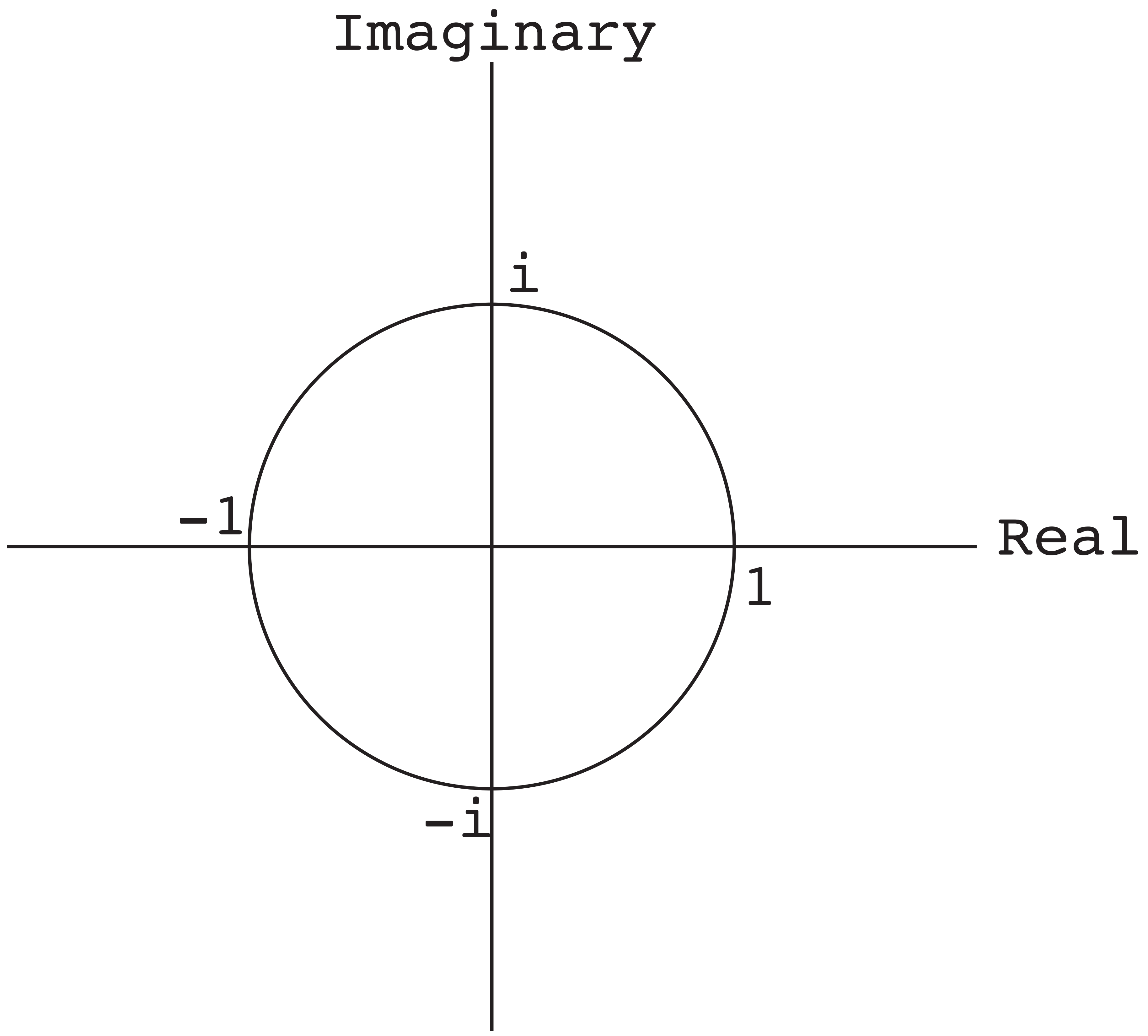
A brief detour…

Complex Numbers

- A complex number is the sum of a real number and an imaginary number:

$$a + bi$$

- Think of as a point on the *complex plane*



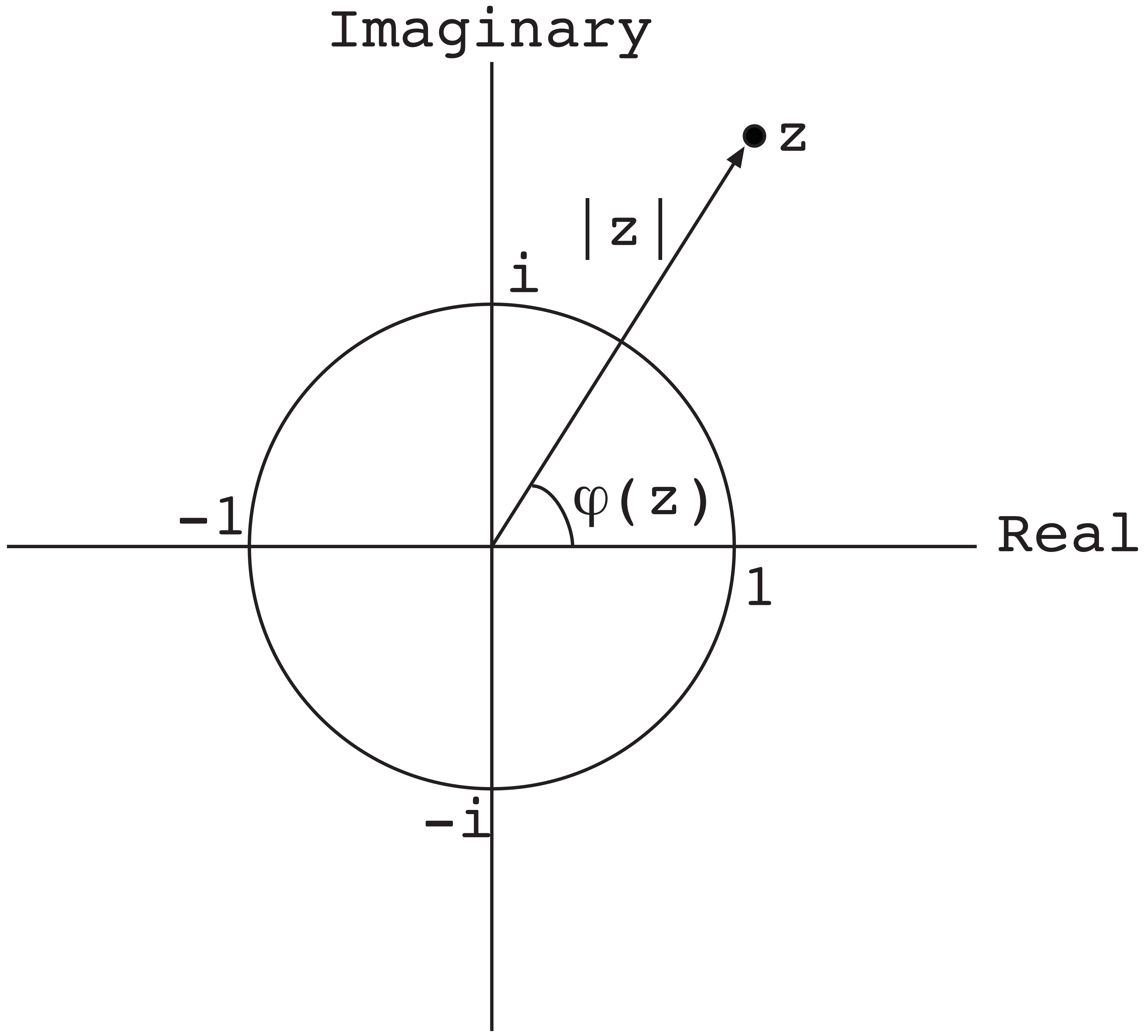
Complex Numbers

- Often useful to think about complex numbers in polar coordinates
- The magnitude is

$$\|a + bi\| = \sqrt{a^2 + b^2}$$

- And the phase (angle) is

$$\phi(a + bi) = \tan^{-1}\left(\frac{b}{a}\right)$$



So why do we care about complex numbers?

Adding Sinusoids

- When you add some amount of a cosine to another amount of a sine of the same frequency, you get a sinusoid of the same frequency

$$a\cos(2\pi ut) + b\sin(2\pi ut)$$

- If you encode the amount of cosine and the amount of sine as a complex number…

$$z = a + bi$$

  - The amplitude of the sinusoid is the magnitude of the complex number
  - The phase of the sinusoid is the phase of the complex number

$$||z|| \ \cos(2\pi ut + \phi(z))$$

Two Ways to Make Sinusoids

- We thus have two ways to make *the same sinusoid*:
  - Mix a cosine and a sine with specific weights
  - Start with a cosine and
    - Stretch it by the square root of the sum of the squares of the sine and cosine weights
    - Shift it by the arctangent of the ratio of the sine weight to the cosine weight

Sinusoidal Weights as Complex Numbers

- Useful to encode the coefficients from projecting onto sinusoidal basis functions as a single complex number

$$c(u) = \int_{-\infty}^{\infty} f(t) \ \cos(2\pi u t) \ dt$$

$$s(u) = \int_{-\infty}^{\infty} f(t) \ \sin(2\pi u t) \ dt$$

$$F(u) = c(u) - i \ s(u)$$

* Note the minus sign here when combining the two, this comes from doing linear algebra with complex quantities

The Fourier Transform

- An encode as a function f(t) as a weighted sum of sines and cosines where the weights are given by a complex-valued function F(u)
- Can think of as an operator
- Written as

$$c(u) = \int_{-\infty}^{\infty} f(t)\ \cos(2\pi ut)\ dt$$

$$s(u) = \int_{-\infty}^{\infty} f(t)\ \sin(2\pi ut)\ dt$$

$$F(u) = \mathcal{F}(f(t))$$

$$F(u) = c(u) - i\ s(u)$$

The Inverse Fourier Transform

- Can invert the transformation to get f(t) from F(u) by simply adding the sines and cosines back up with the respective weights
- Can also think of as an operator
- Written as

$$f(t) = \mathcal{F}^{-1}(F(u))$$

$$F(u) = a(u) + i\ b(u)$$

$$\begin{aligned} f(t) \quad &= \quad \int_{-\infty}^{\infty} a(u)\ \cos(2\pi ut)\ du \\ &+ \quad \int_{-\infty}^{\infty} b(u)\ \sin(2\pi ut)\ du \end{aligned}$$

The Discrete Fourier Transform

- What about a finite-length, sampled signal?
- The Fourier Transform assumes an infinite-length signal (impossible to work with on computers)
- What should we assume about the signal before and after we record it?
- Since we're decomposing it into a sum of periodic functions, *let's assume it's one period of an infinite periodic function*

Period Functions

- For a periodic signal with period N units, all of the underlying frequencies must also repeat over the period N
- So, each component frequency must be a multiple of the frequency of the periodic signal itself:
- There are no more than N components for a signal with period N samples!

$$\frac{0}{N}, \frac{1}{N}, \frac{2}{N}, \frac{3}{N} \cdots \frac{N-1}{N}$$

The Discrete Fourier Transform

- Same idea as the Fourier Transform
- For a finite with N samples:
  - N discrete frequencies *u/N*
  - Sum over the N discrete samples
- What would the code look like?
- What is the complexity?

$$c[u] = \frac{1}{N} \sum_{u=0}^{N-1} f[t] \ \cos(2\pi ut/N)$$

$$s[u] = \frac{1}{N} \sum_{u=0}^{N-1} f[t] \ \sin(2\pi ut/N)$$

$$F[u] = c[u] - i \ s[u]$$

The Fast Fourier Transform

- The Fast Fourier Transform does exactly the equivalent mathematically
- Divide-and-conquer algorithm provides greater efficiency

DFT	$O(N^2)$
FFT	$O(N \log N)$

Using the FFT

- Pass in an array of length N
- Returns back an array of length N <u>of type complex</u>
  - The <u>real part</u> of each number is how much of a <u>cosine</u> of that frequency there is
  - The <u>imaginary part</u> of each number is how much of a <u>sine</u> of that frequency there is

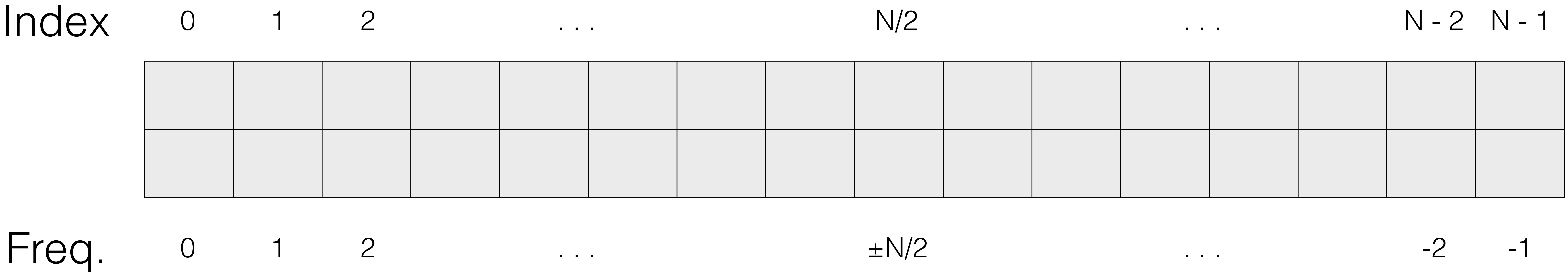
Interpreting the Complex Numbers

- Can think of in Cartesian form:
  - The <u>real part</u> of each number is <u>how much of a cosine</u> of that frequency
  - The <u>imaginary part</u> of each number is <u>how much of a sine</u> of that frequency
- Or in polar form:
  - The <u>magnitude</u> of each number is <u>how much</u> of that frequency there signal
  - The <u>phase</u> of each number is the <u>relative shift</u> (from a cosine) of each sinusoid

Negative Frequencies

- Because of their repetitive nature
  - cosines are symmetric $\cos(-2\pi u/N) = \cos(2\pi u/N)$
  - sines are antisymmetric $\sin(-2\pi u/N) = -\sin(2\pi u/N)$
  - discrete ones repeat modulo N $N - u \equiv -u \mod N$

Storage of the DFT Results



Because of symmetry, the last half of the array is a mirrored (negated) copy of the first half

$$\cos(-2\pi u/N) = \cos(2\pi u/N)$$

$$\sin(-2\pi u/N) = -\sin(2\pi u/N)$$

Implications

- Because of symmetry
  - Last half of the real part is a mirrored copy of the first half
  - Last half of the imaginary part is a negated mirrored copy
- There are really only N basis functions, as there should be, not 2N
- Only really computing frequencies up to N/2
  - Twice the number of samples are there are frequencies
  - “Sample at twice the highest frequency in the signal…”
  - This is the same as Shannon’s sampling theorem!

Useful Python Functions

- **`np.fft.fft`** - forward FFT (complex array in, complex array out)
- **`np.fft.ifft`** - inverse FFT (complex array in, complex array out)
- **`np.absolute`** - returns the magnitude of a complex number (or the absolute value of a real one)
- Normal mathematical operators (+, -, *, /) work on complex numbers as well as integer, floating point ones
- Tip: a real number is a complex one with imaginary part equal to zero

Why Are We Doing This Again?

- The Fourier Transform of a signal lets you analyze the mix of frequencies in it
- Can manipulate the transformed signal and then transform it back! (that’s the topic for the next class)

Coming up...

- Examples and properties
- Filtering in the frequency domain
- The Convolution Theorem (wait, what? convolution?)
- 2-D FFT and image filtering in the frequency domain