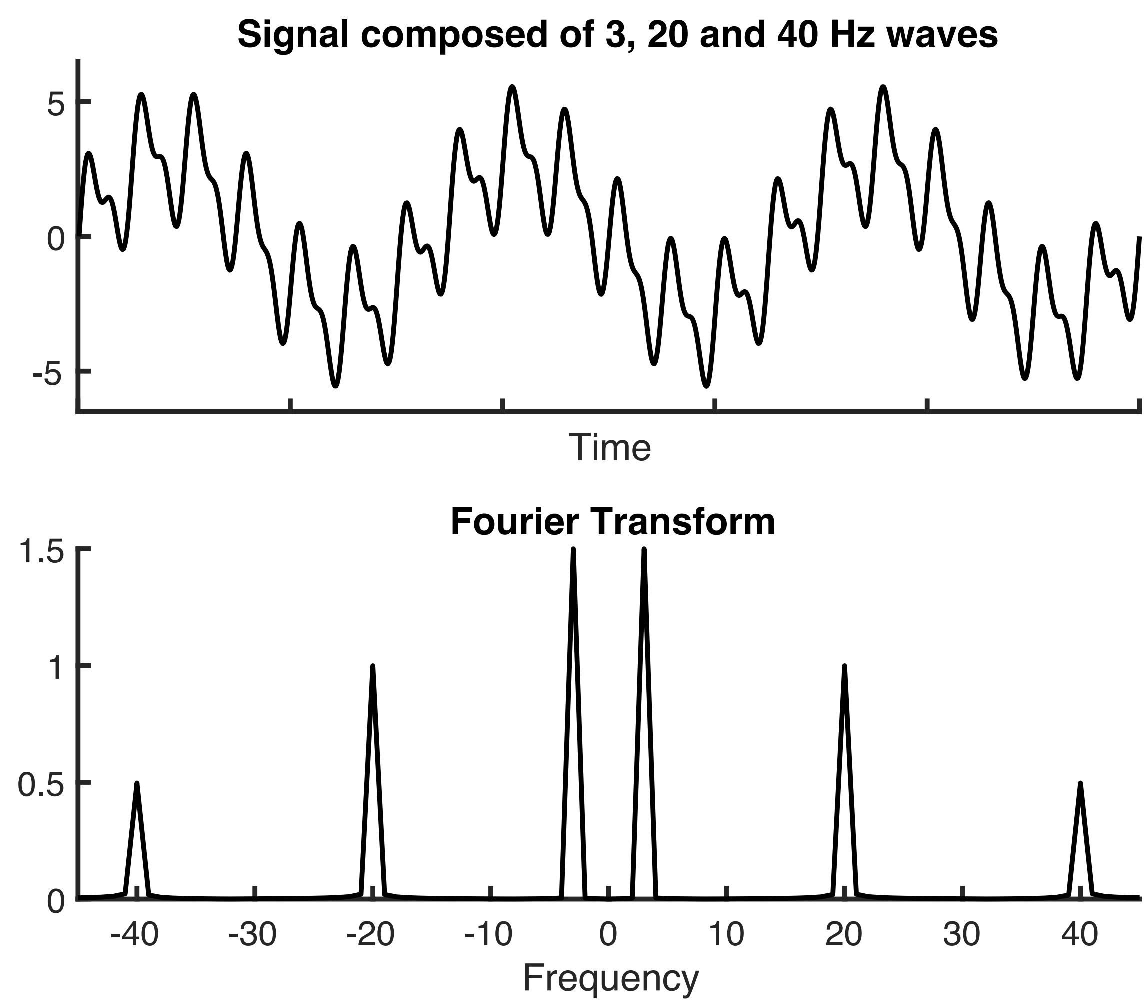
Fourier Transform

$$F(\omega) = \int_{-\infty}^{\infty} \underbrace{f(t)}_{\text{your timeseries signal}} \underbrace{e^{-i\omega t}}_{\text{sinusoid (more about this later)}} dt$$

Fourier transform (FT) is the product of a **signal** with a **sinusoid** at a particular frequency, summed (integrated) across all time…should sound familiar

Fourier Transform

FT of the sum (same as
sum of the FTs)



Fourier Transform

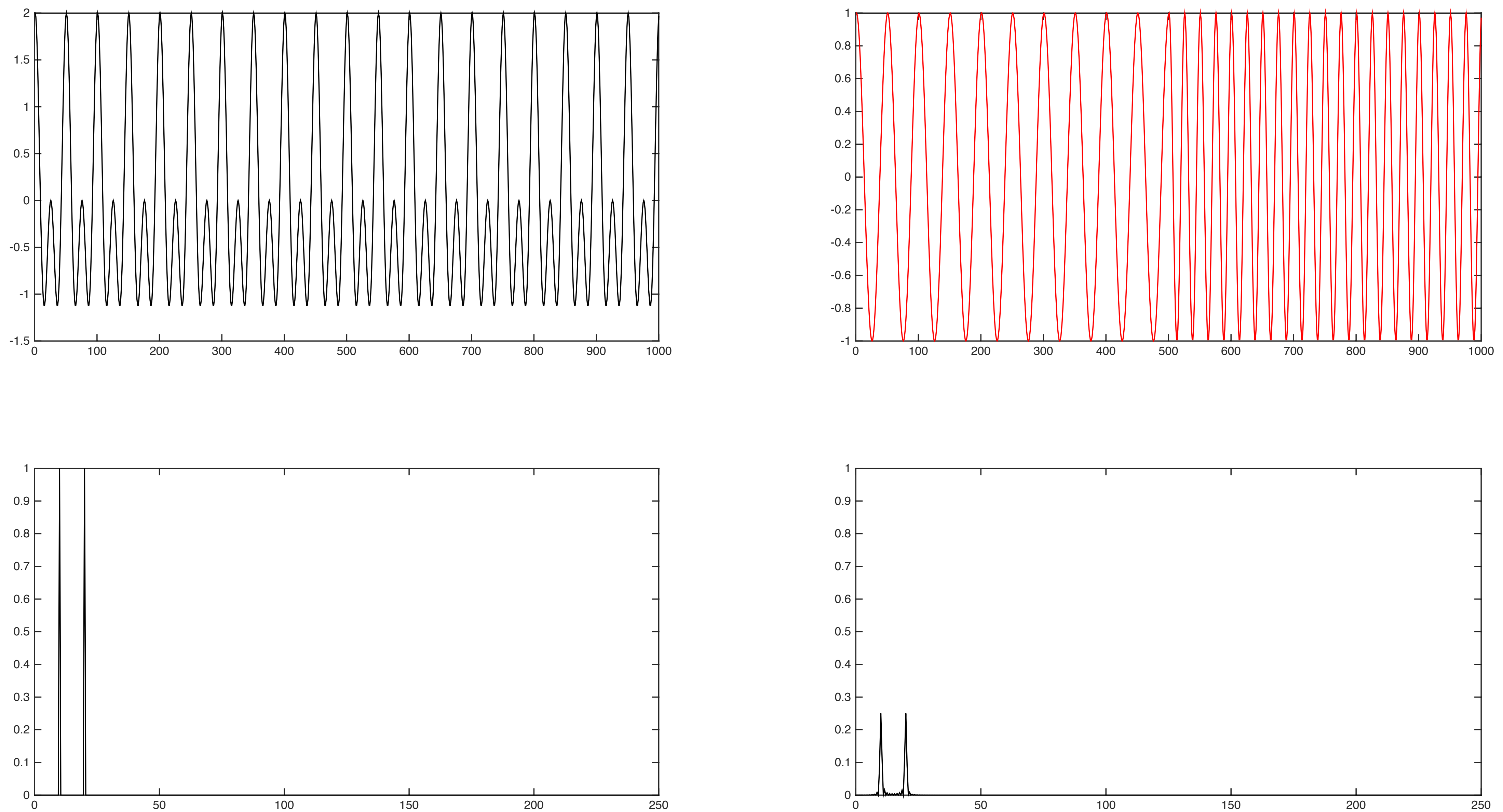
$$F(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt$$

your timeseries signal

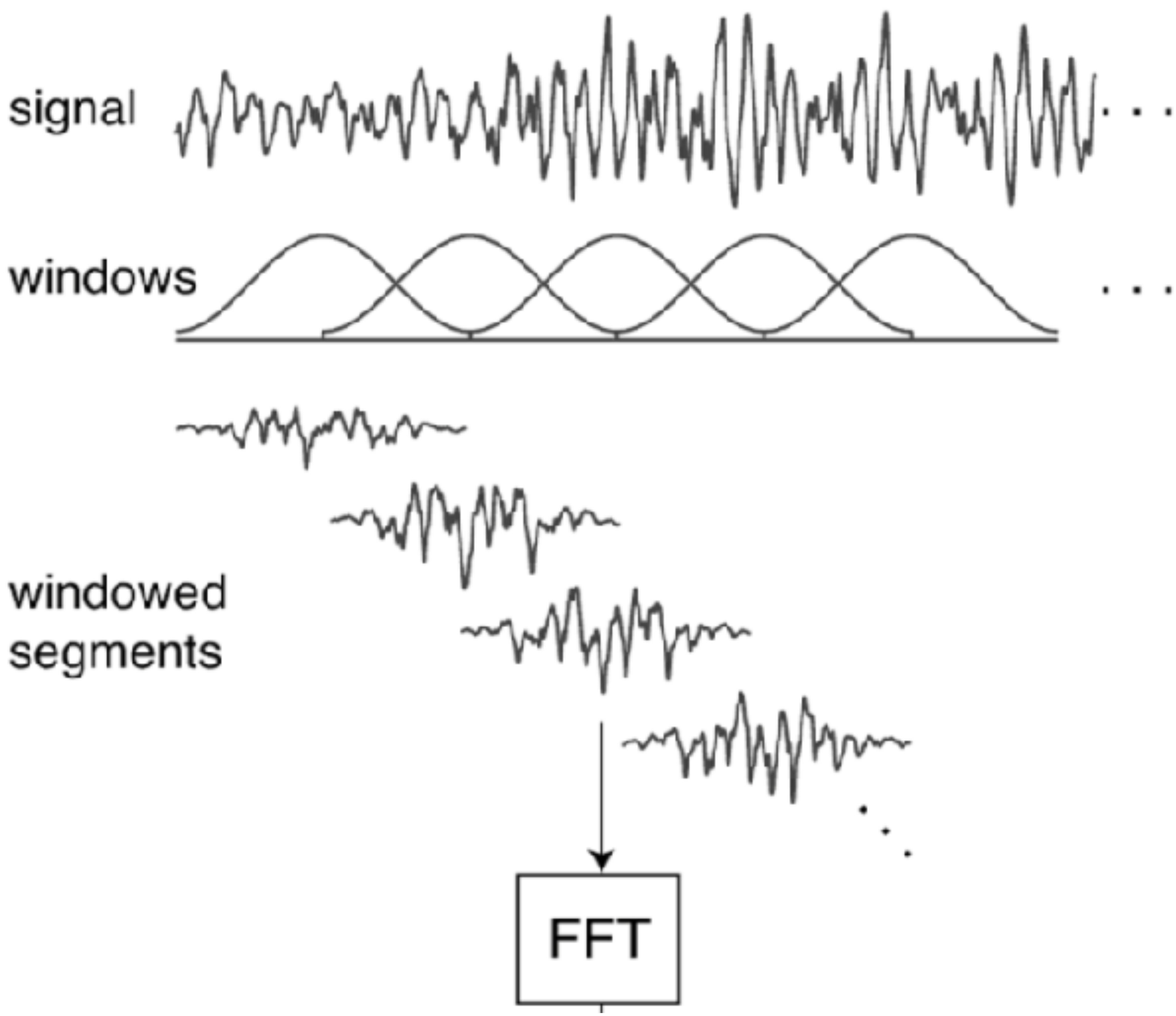
sinusoid (more about this later)

Fourier transform (FT) is the product of a **signal** with a **sinusoid** at a particular frequency, summed (integrated) across all time…should sound familiar

Limitation of FFT: If signal composition changes over time (**stationarity**)

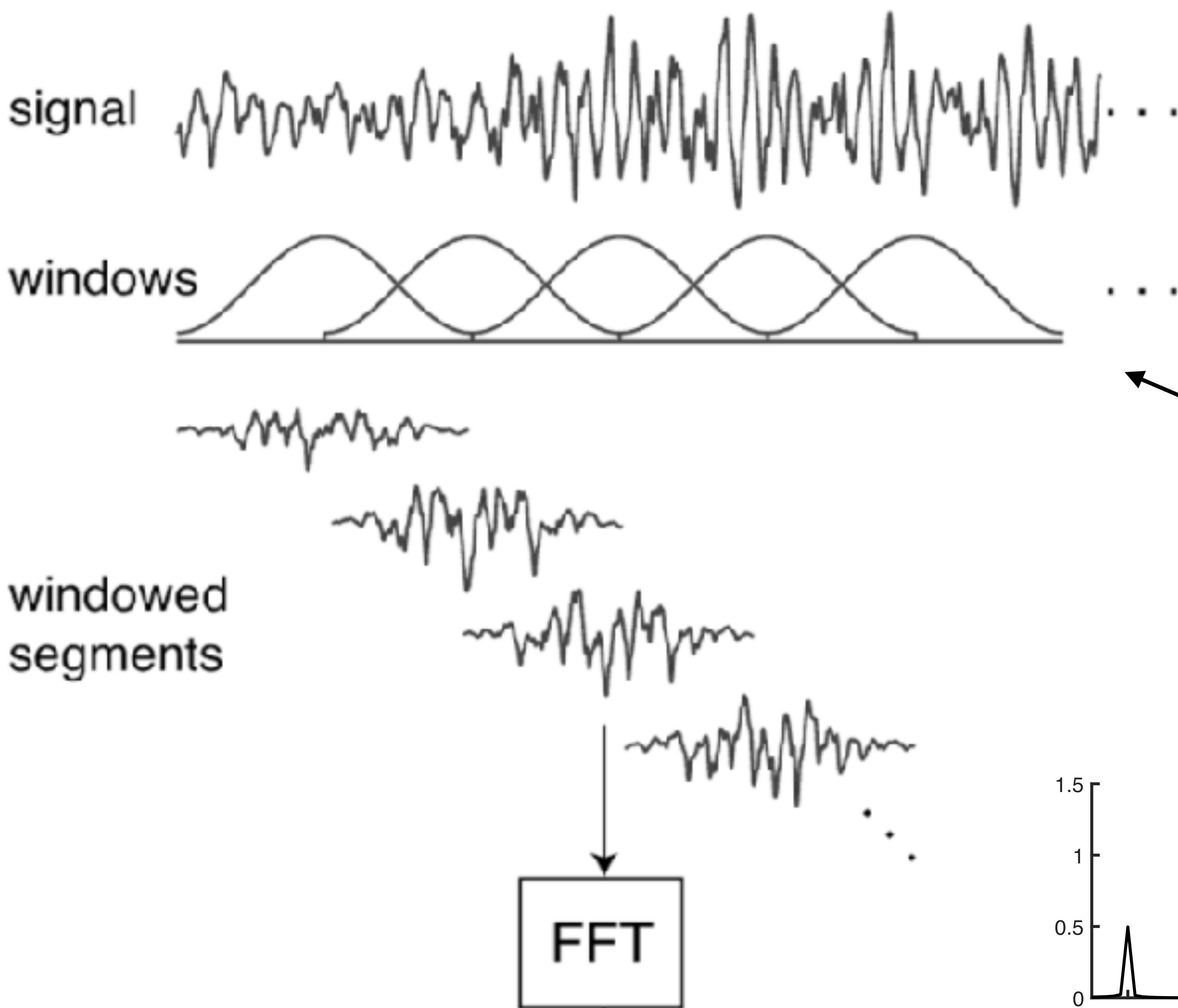


Short-time Fourier Transform



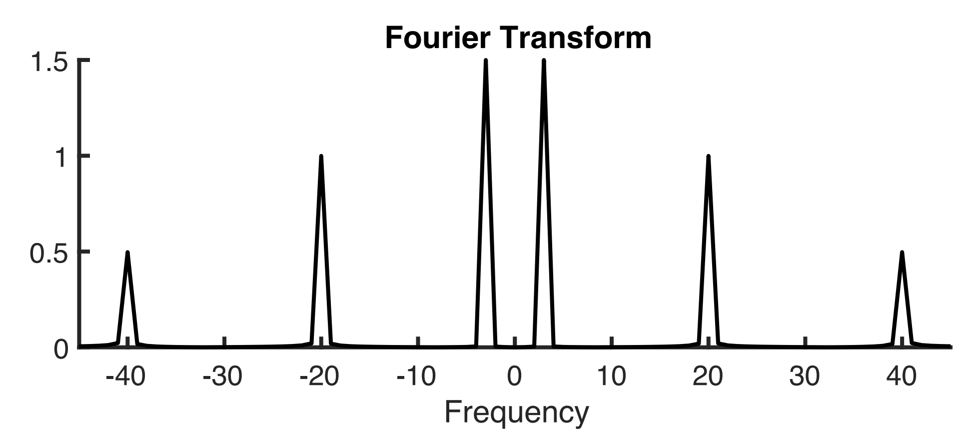
Sethares (2007)

Short-time Fourier Transform



Why non-rectangular windows?

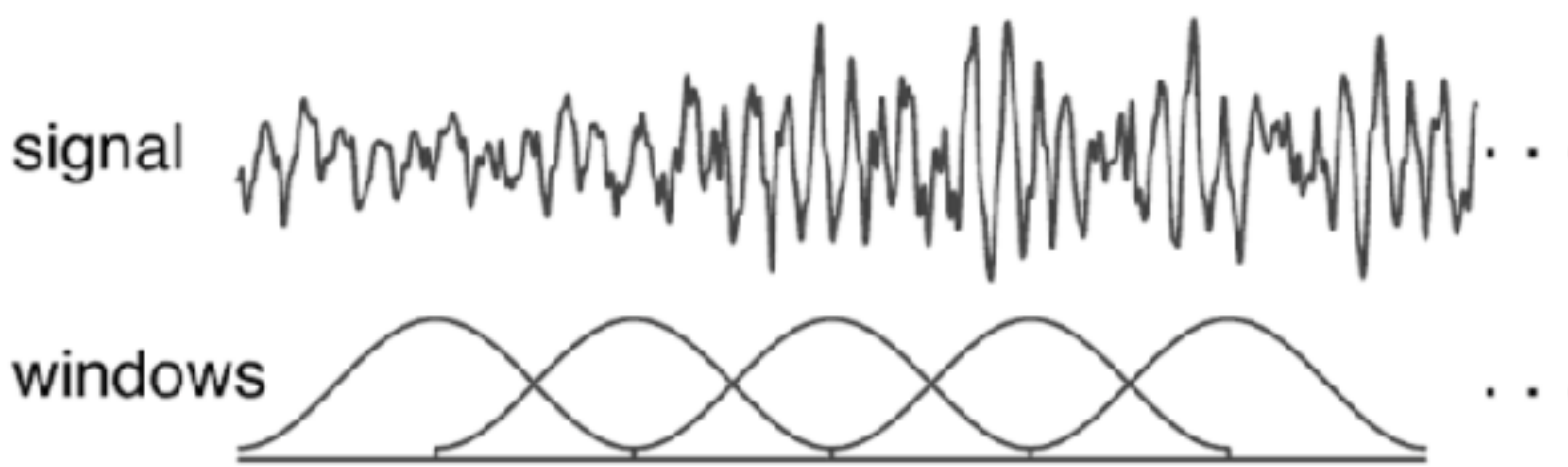
spectral leakage



Sethares (2007)

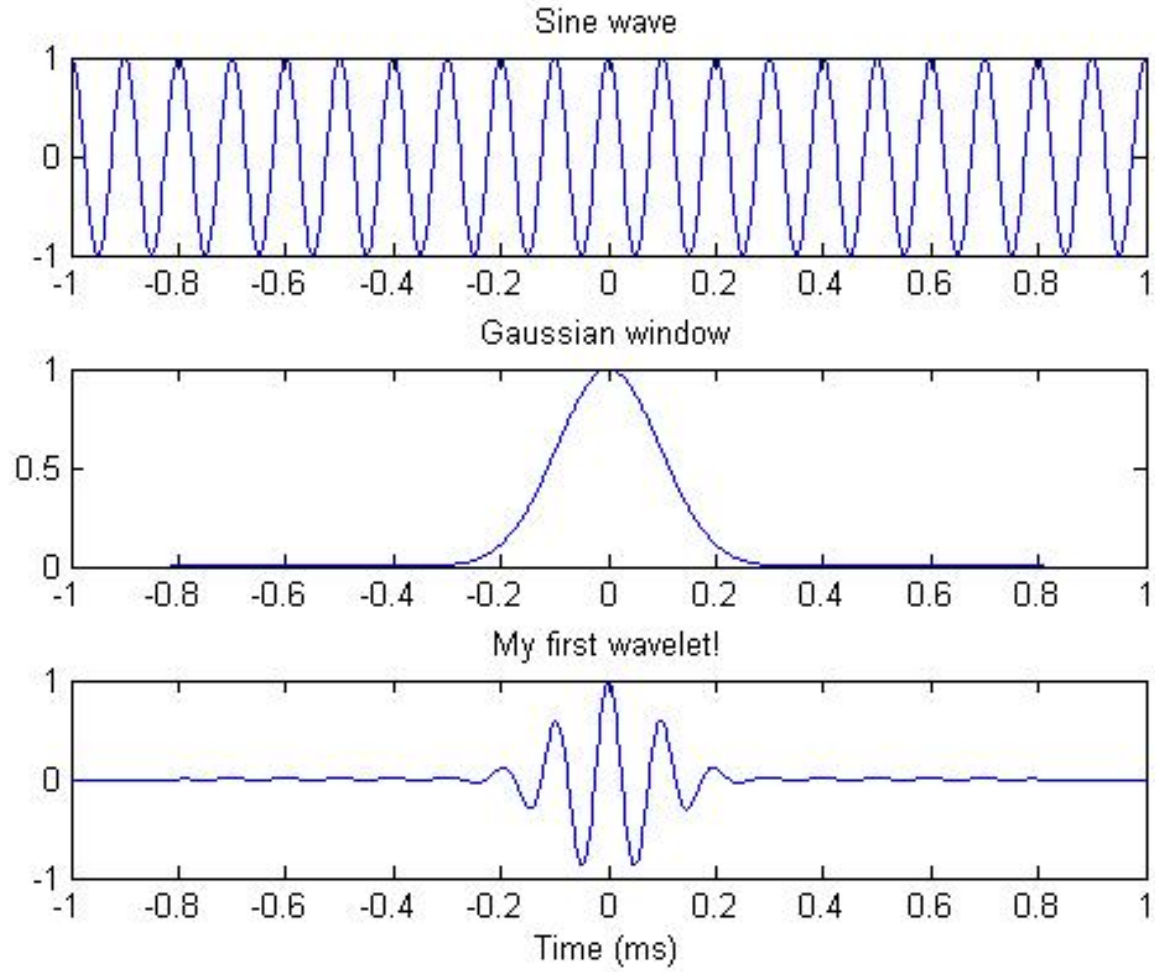
Short-time Fourier Transform

- Choice of window size affects signal-to-noise—longer windows needed to capture enough cycles of low-frequency oscillations
- If using longer windows, you have reduced sensitivity to changes in high-frequency oscillations
- Using longer windows reduces your ability to resolve when oscillatory changes occur
- Morlet wavelets address some of these issues



Wavelets

- wavelet = small wave
- take a sinusoid, multiply it by a Gaussian that 'windows' it to be (mostly) limited to brief time period



Wavelets

$$h[n,k] = \underbrace{Ae^{-\frac{(n-\mu)^2}{2\sigma^2}}}_{\text{Gaussian}}\ \underbrace{e^{\frac{-i2\pi kn}{N}}}_{\text{sinusoid}}$$

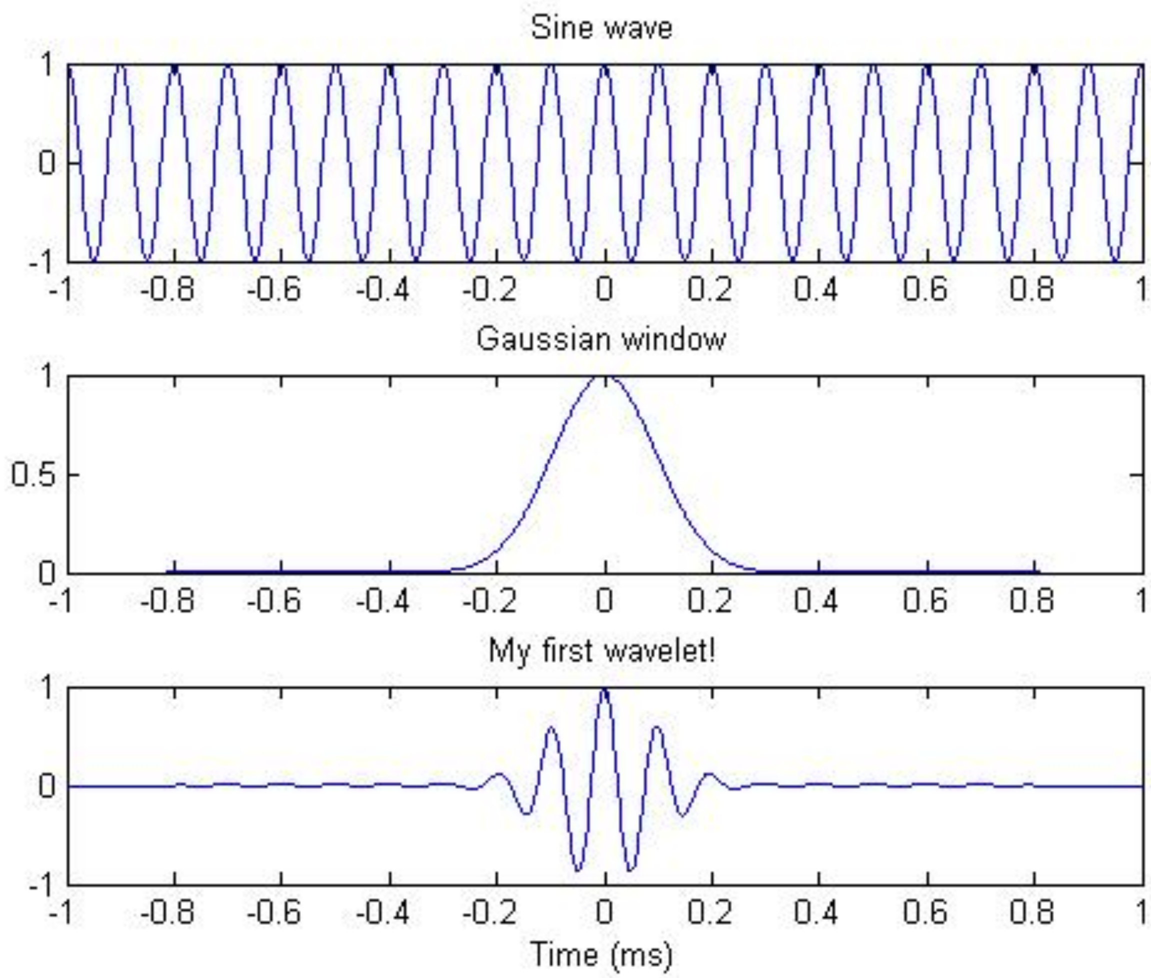
Normalization factor
for the Gaussian

$$A = \frac{1}{(\sigma\sqrt{\pi})^{1/2}}$$

$$\mu = mean$$
$$\sigma = SD$$

Wavelets

$$h[n,k] = Ae^{-\frac{(n-\mu)^2}{2\sigma^2}} e^{\frac{-i2\pi kn}{N}}$$

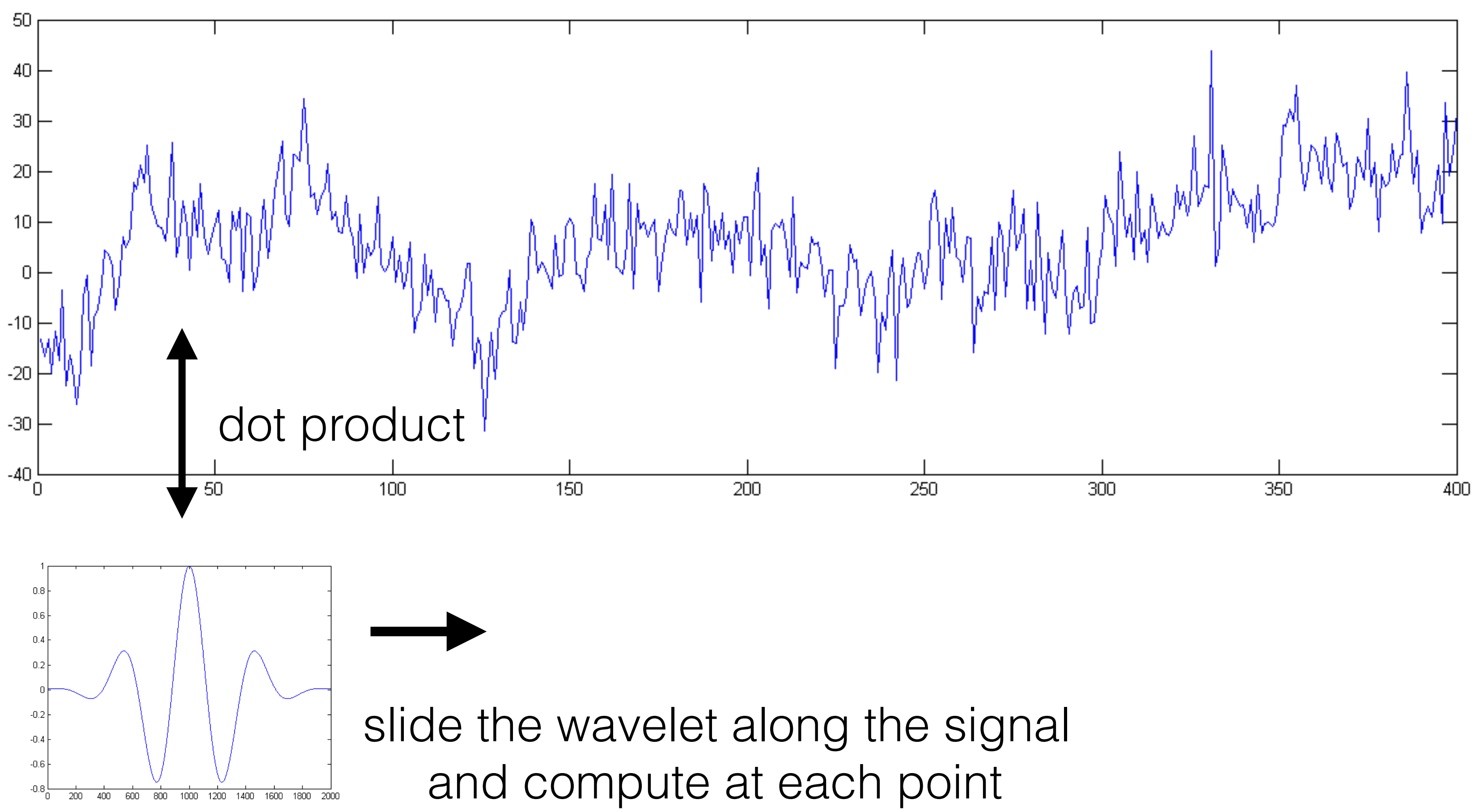


wave number specifies the width of the Gaussian which determines the width of the wavelet

$$\sigma = \frac{c_n}{2\pi k}$$

Wavelet method

- Wavelet method:
  - Create a wavelet for a frequency of interest
  - Compute the dot product between your signal and your wavelet
  - Slide the wavelet along the signal
- Sliding dot product is called **convolution**

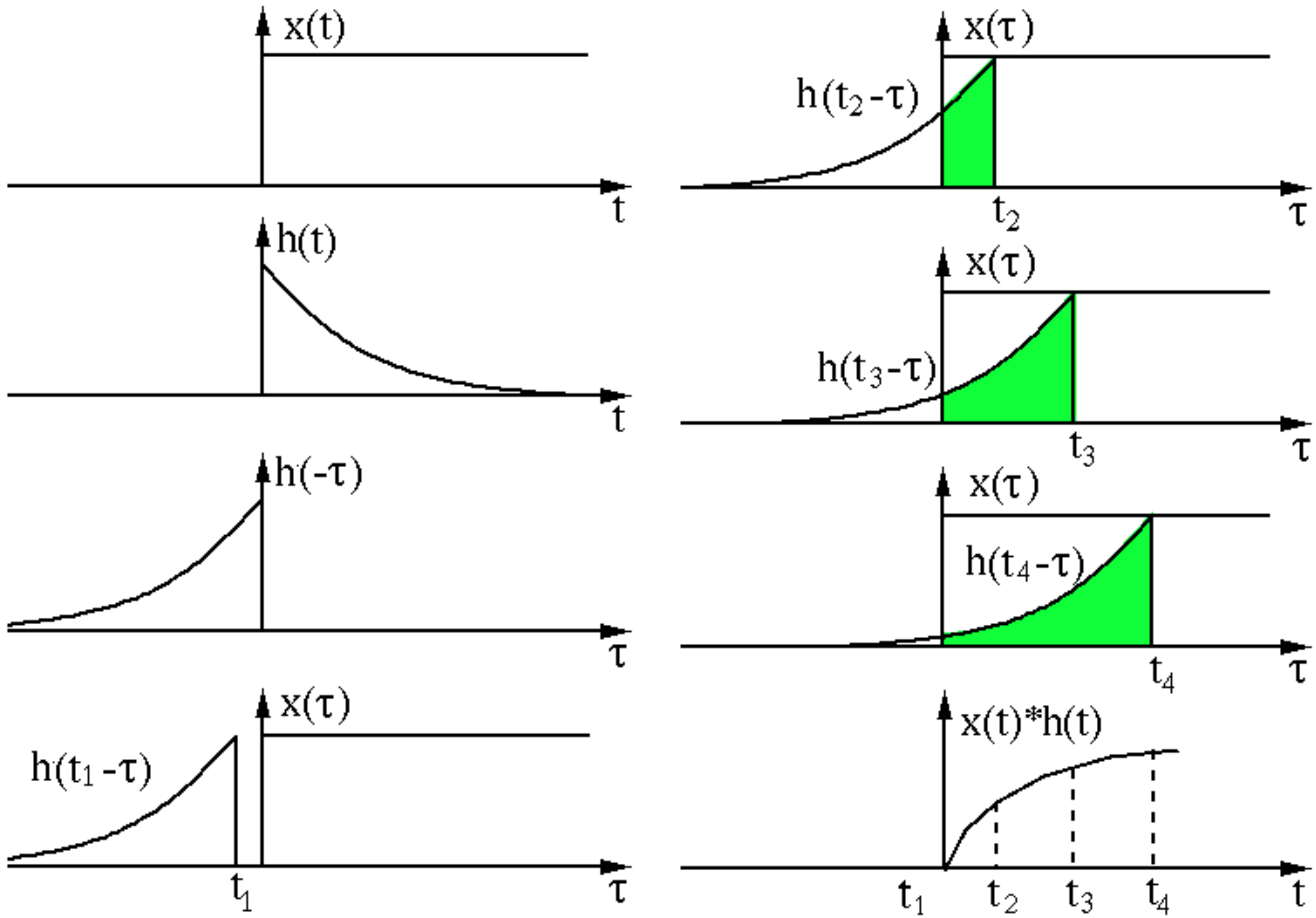


Convolution

- Intuitive ways to think about convolution:
  - Computes the similarity between two signals as a function of time
  - Computes the effect of ‘applying’ one signal to another
- x[n] = signal
- h[n] = kernel

$$x[n] * h[n] = \sum_{m=0}^{N-1} x[m]h[n-m]$$

Convolution



Convolution Theorem

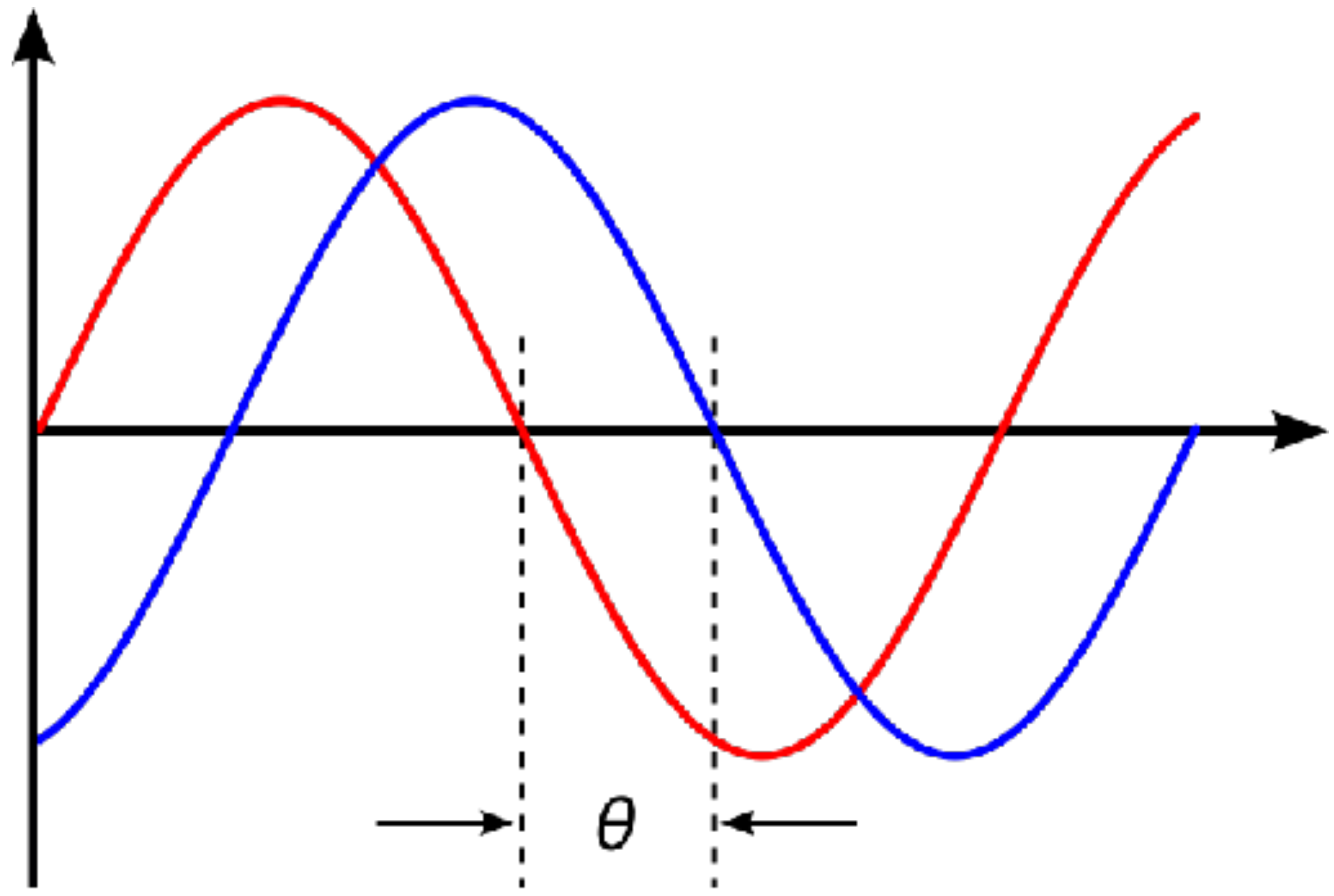
- Convolving two signals in the **time domain** is equivalent to multiplying them in the **frequency domain** and taking the inverse Fourier transform

- Because of efficient algorithms for computing the forward and inverse Fourier transforms, performing three Fourier transforms and one multiplication is much faster than convolution in the time domain, especially for long signals

$$x[n] * h[n] = \mathcal{F}^{-1}\{\mathcal{F}\{x[n]\} \cdot \mathcal{F}\{h[n]\}\}$$

Amplitude and Phase

- A signal can be reconstructed from its Fourier transform
- Only if you have amplitude AND phase information
- Complex sinusoids simplify the mathematics of extracting both amplitude and phase information

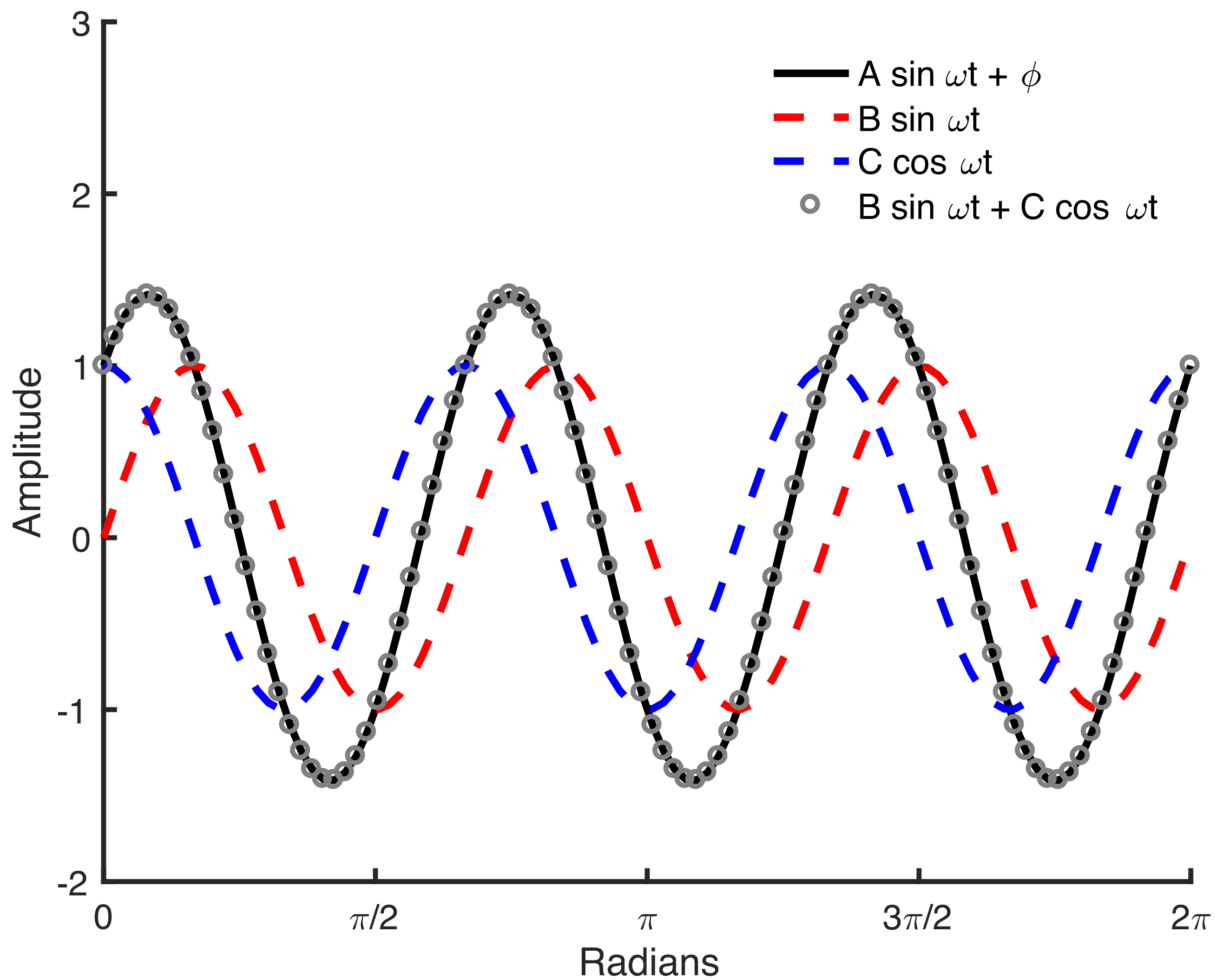


Complex sinusoid

- complex sinusoid is the sum of a real and imaginary part
- cos = real part, sin = imaginary part

$$x(t, f) = M[\cos(2\pi f t) + i \sin(2\pi f t)]$$

Sinusoids can be written as the sum of a cosine and sine



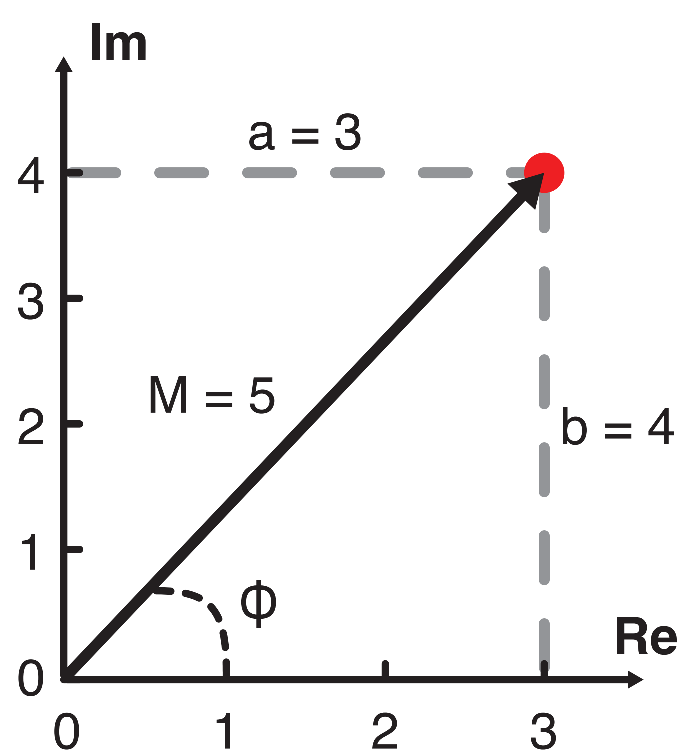
$$M \sin \omega t + \phi = a \cos \omega t + b \sin \omega t$$

$M = \sqrt{b^2 + a^2}$ and $\phi = \arctan(\frac{b}{a})$

Complex sinusoid

- Complex numbers provide a convenient way of representing two quantities in a single number
- a + bi

3 + 4i



$$M = \sqrt{Re[A]^2 + Im[A]^2} = \sqrt{a^2 + b^2}$$

$$\phi = \arctan(\frac{b}{a})$$

Euler's formula

- a complex sinusoid can also be written as a complex exponential

$$x(t, f) = M[\cos(2\pi f t) + i \sin(2\pi f t)]$$

$$e^{ix} = \cos(x) + i \sin(x)$$

greatly simplifies the math of dealing with sinusoids