

# CS 583 – Computational Audio -- Fall, 2021

Wayne Snyder  
Computer Science Department  
Boston University

---

Today: [Warmup to Fourier Analysis: Fundamental Frequency Detection](#)

What happens when we multiply sine waves?

Correlation of two signals

Autocorrelation of a signal with itself

Using Autocorrelation for F0 Detection

Other methods (Zero-crossing rate, histograms)



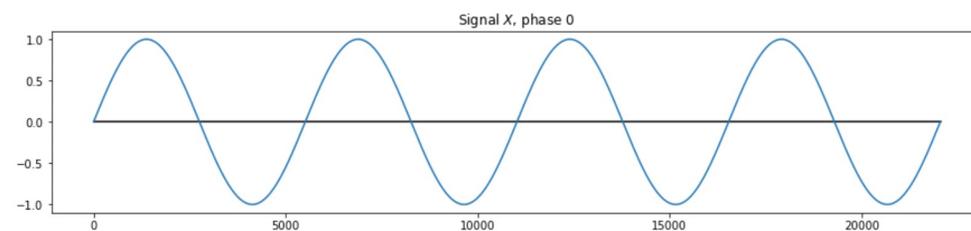
# Multiplying Sine Waves



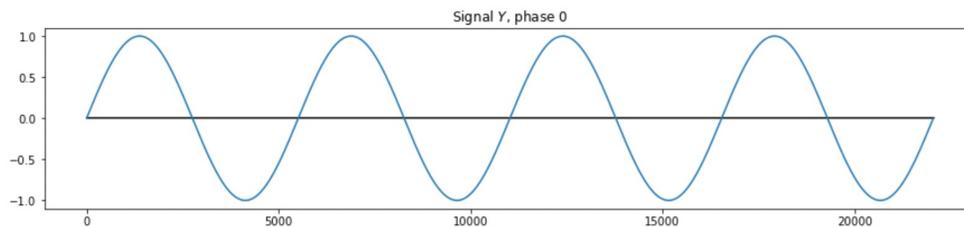
Computer Science

One of the fundamental operations in audio analysis is to multiply two sine waves to see how similar they are. This can be used to analyze the wave's characteristics. Let's understand the fundamental idea. First let's analyze phase.....

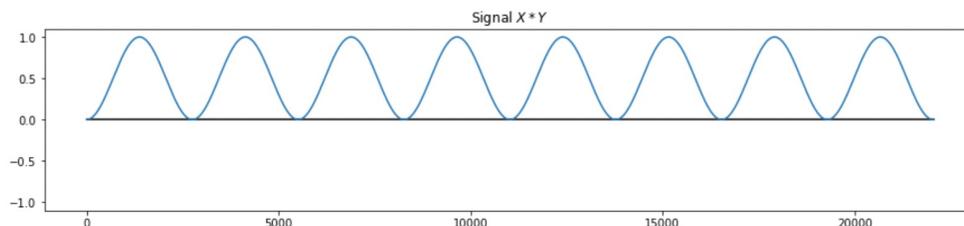
Suppose the two waves have identical phase (we're squaring the wave!):



mean(X) = 0.0



mean(Y) = 0.0



mean(X\*Y) = 0.5

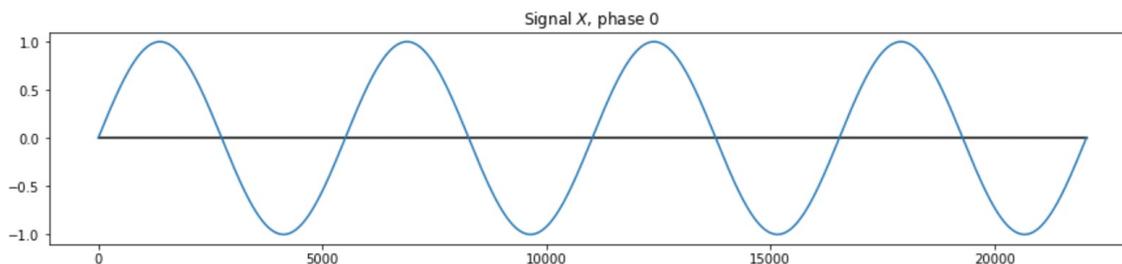
Notice the mean value of the product wave... 0.5.

# Multiplying Sine Waves

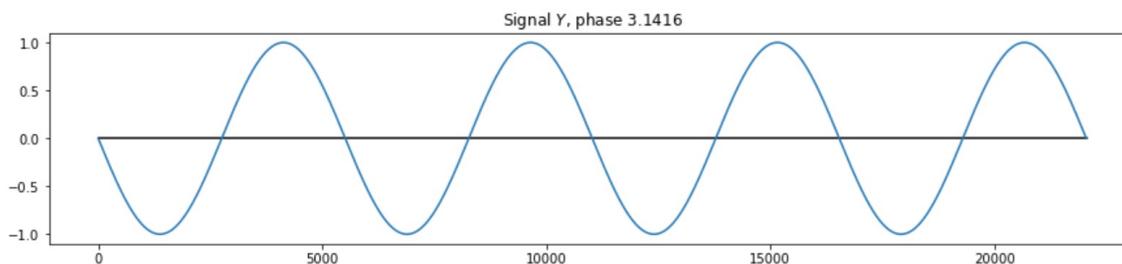


Computer Science

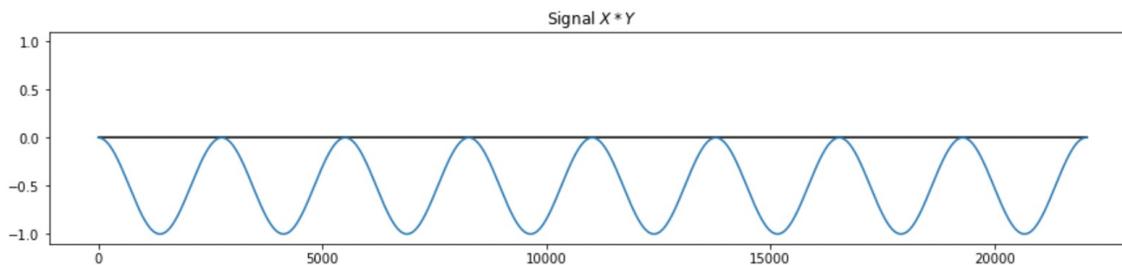
Suppose we do the same thing to two waves "out of phase" with phase difference  $2\pi$ :



`mean(X) = 0.0`



`mean(Y) = -0.0`



`mean(X*Y) = -0.5`

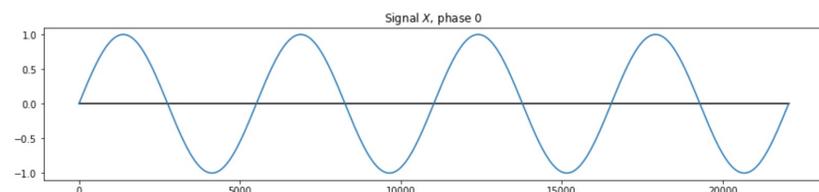
Mean( $X*Y$ ) =  
**-0.5.**

# Multiplying Sine Waves

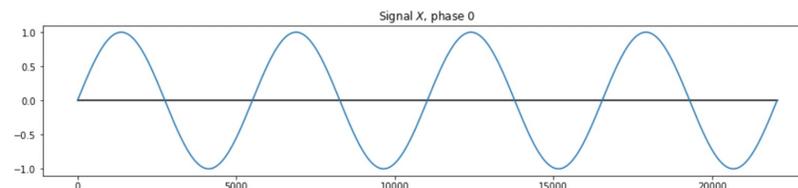


Computer Science

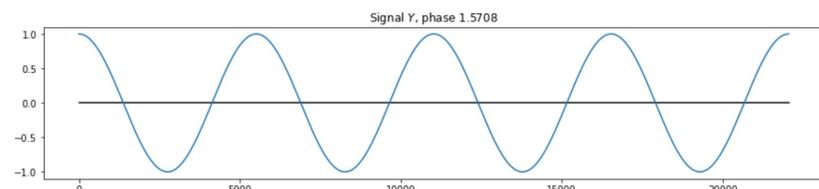
When the phases are off by  $\pm \pi/2$ , we get a mean of 0.0:



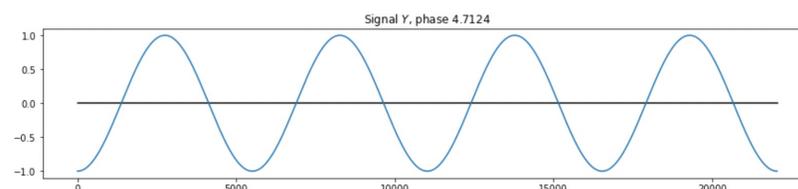
mean(X) = 0.0



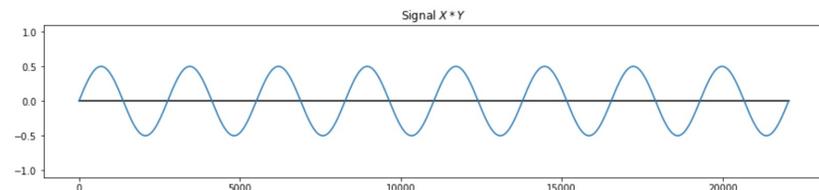
mean(X) = 0.0



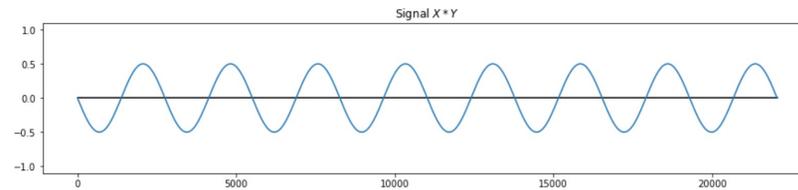
mean(Y) = -0.0



mean(Y) = 0.0



mean(X\*Y) = 0.0



mean(X\*Y) = -0.0

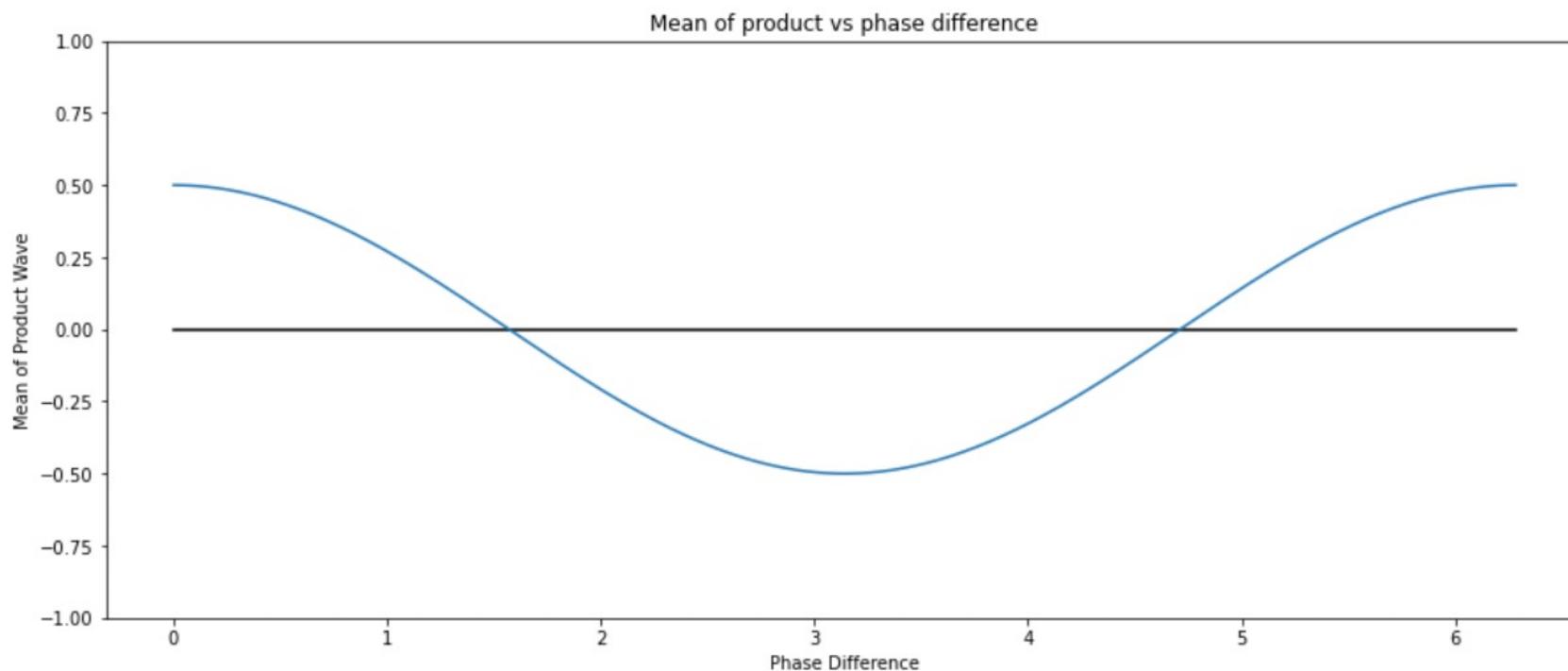
**Punchline: When we multiply two sine waves of the same frequency, we get another sine wave of twice the frequency, possibly shifted on the Y axis.**

# Multiplying Sine Waves



Computer Science

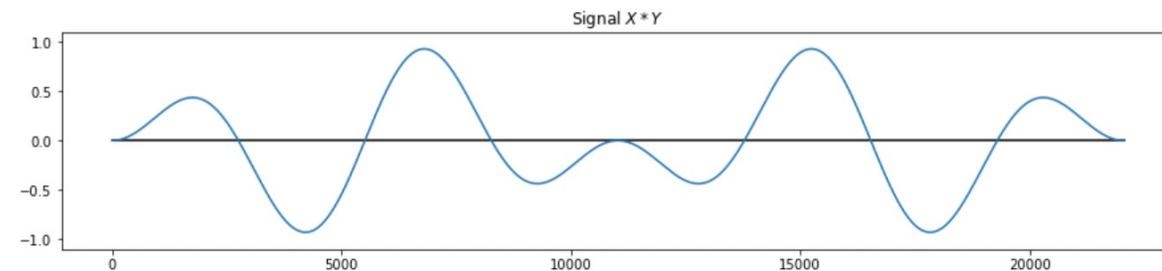
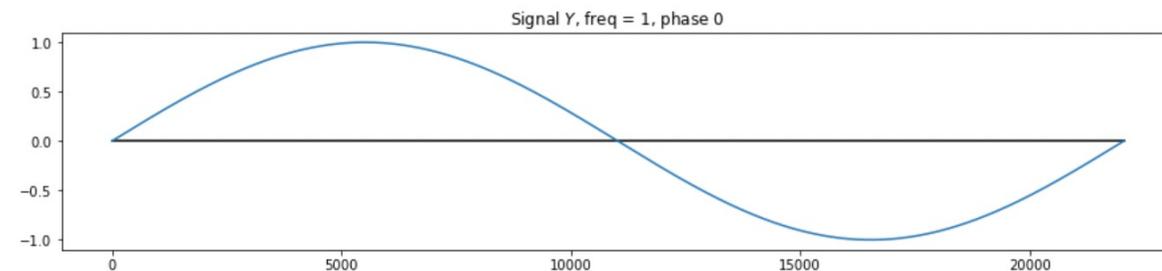
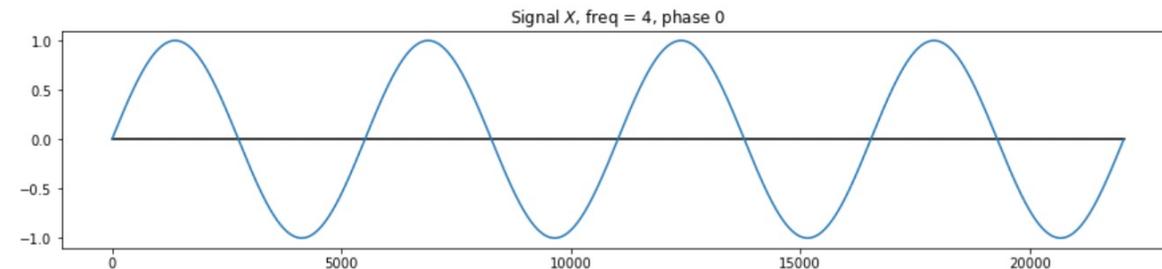
Summary: When we graph the mean of the product wave vs the phase difference, we get a cosine wave of amplitude 0.5....



The mean value tells us how “in phase” the two waves are!

# Multiplying Sine Waves

What happens when we change the **frequency**?



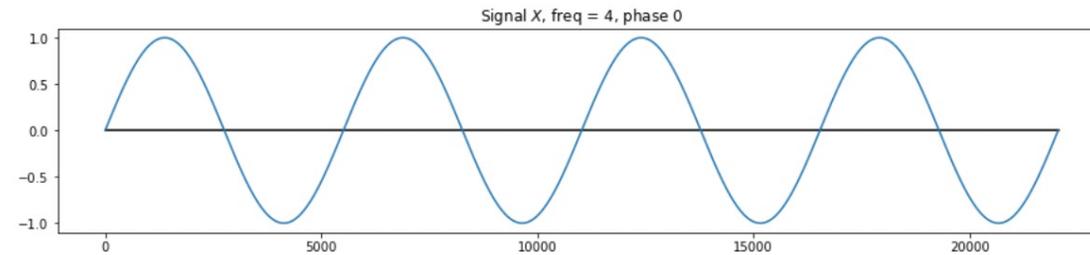
Mean value of  
the product  
wave = 0.0

# Multiplying Sine Waves

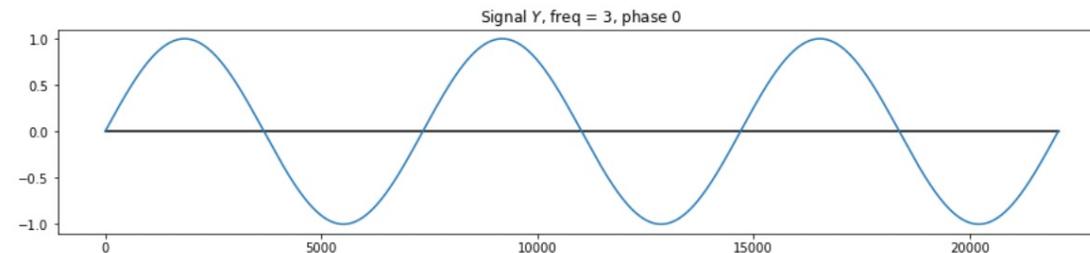


Computer Science

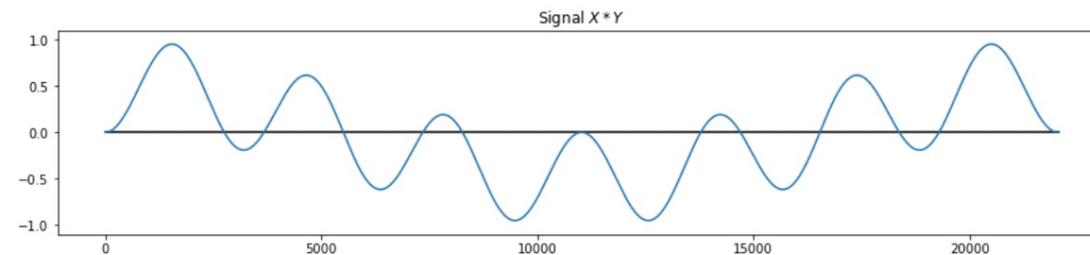
What happens when we change the frequency?



mean(X) = 0.0



mean(Y) = -0.0

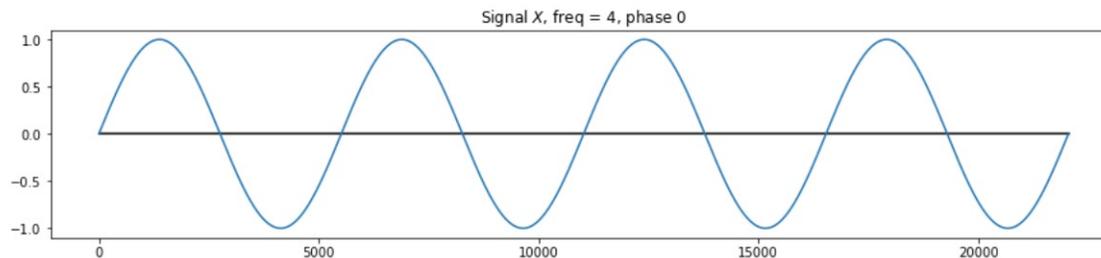


mean(X\*Y) = 0.0

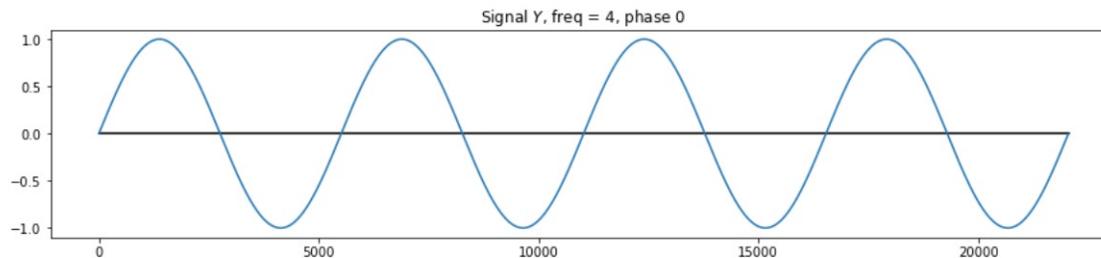
Mean value of  
the product  
wave = 0.0

# Multiplying Sine Waves

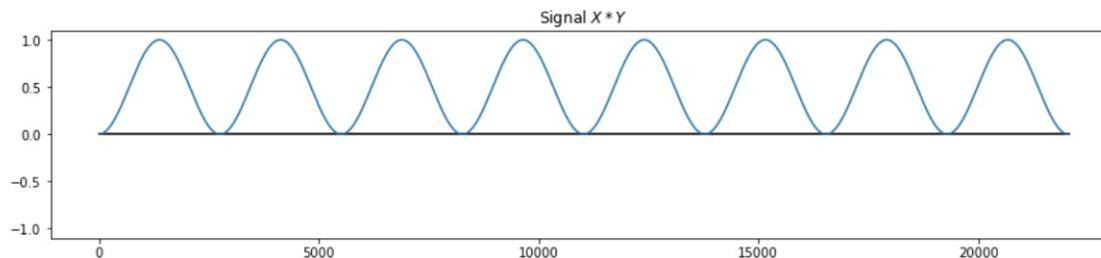
What happens when we change the frequency?



$\text{mean}(X) = 0.0$



$\text{mean}(Y) = 0.0$



$\text{mean}(X*Y) = 0.5$

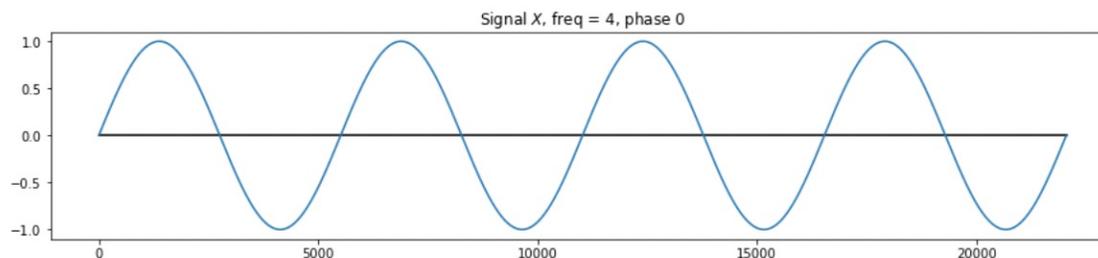
Mean value of  
the product  
wave = 0.5

# Multiplying Sine Waves

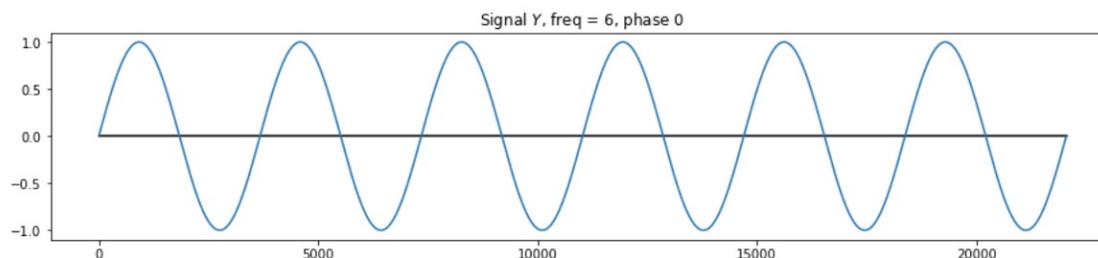


Computer Science

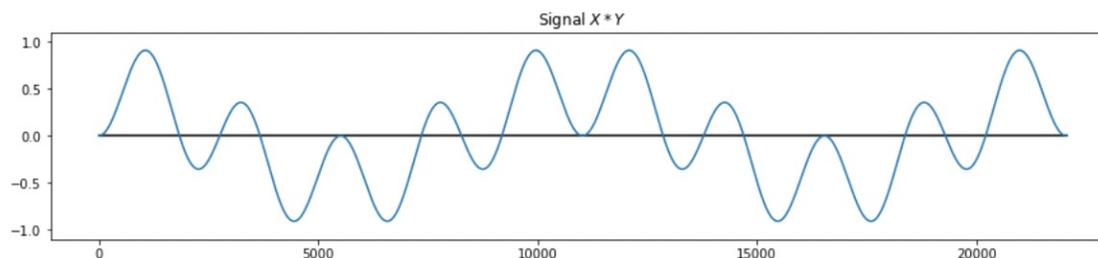
What happens when we change the frequency?



mean(X) = 0.0



mean(Y) = 0.0

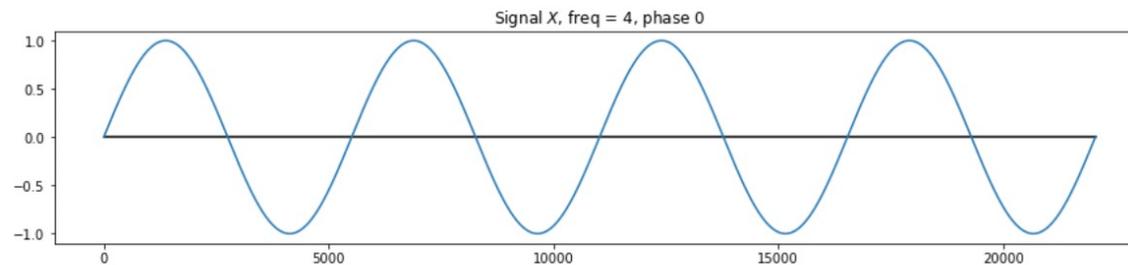


mean(X\*Y) = 0.0

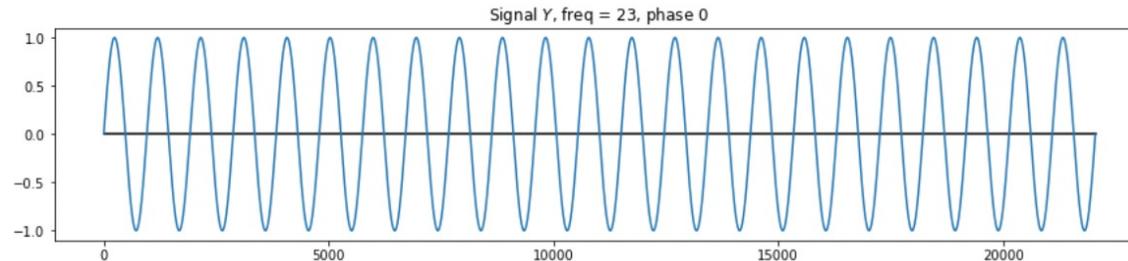
Mean value of  
the product  
wave = 0.0

# Multiplying Sine Waves

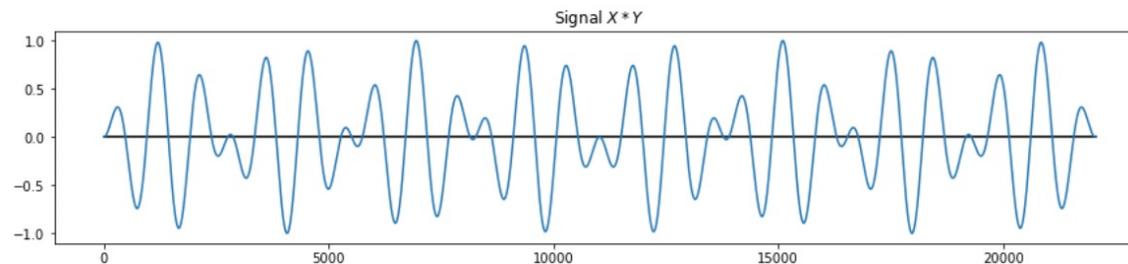
What happens when we change the frequency?



mean(X) = 0.0



mean(Y) = 0.0



mean(X\*Y) = -0.0

Mean value of  
the product  
wave = 0.0

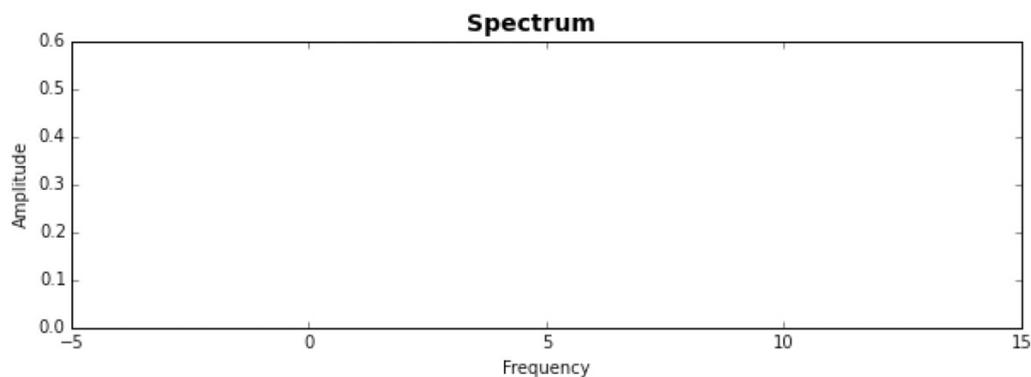
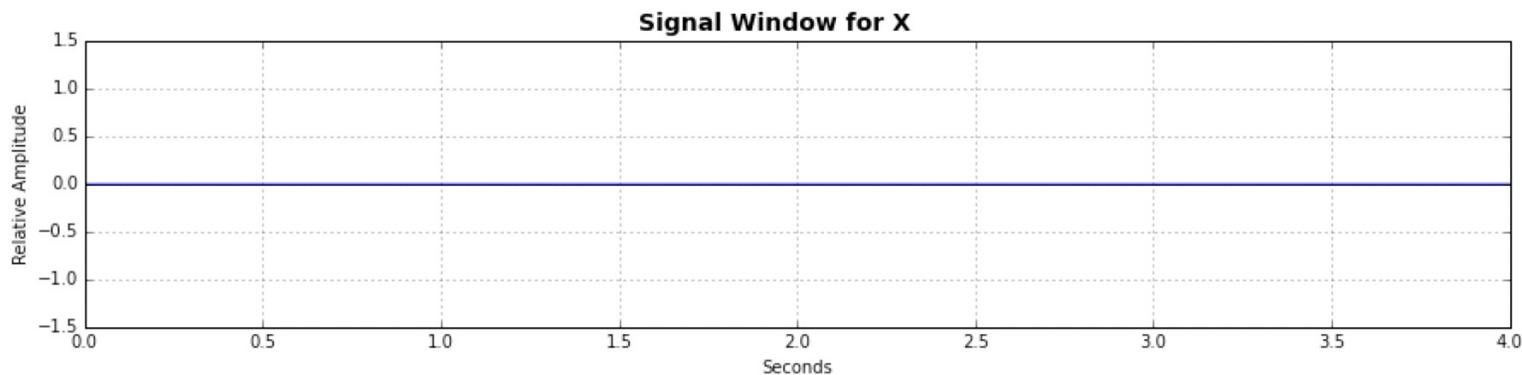
# Multiplying Sine Waves



Computer Science

Here is what you get when you vary the frequency differential:

```
In [15]: showProducts()  
Product of 5 Hz and 0 Hz
```



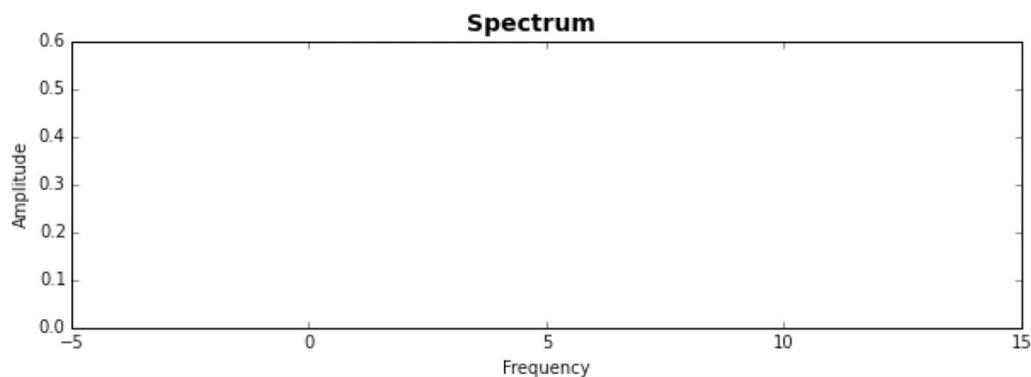
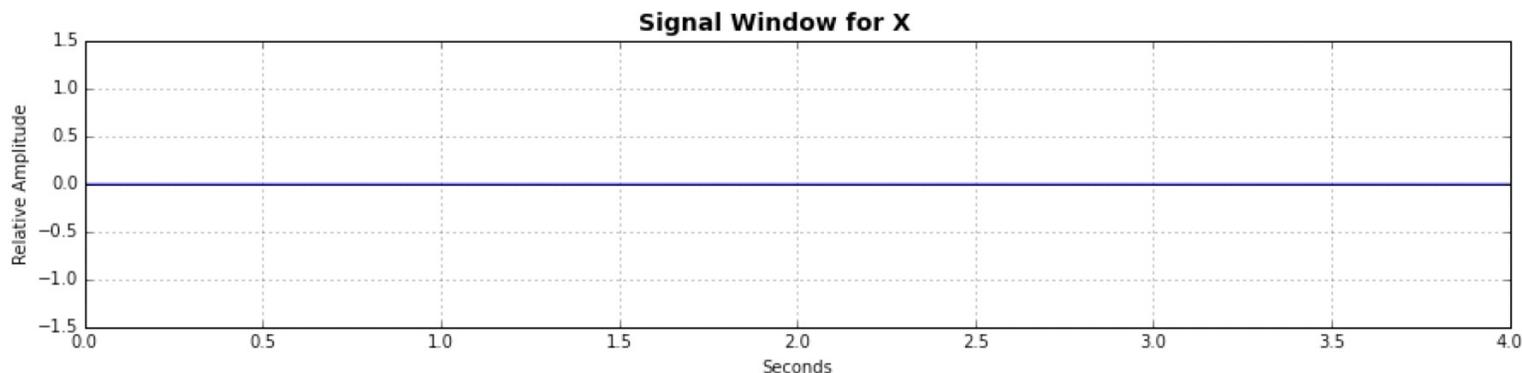
# Multiplying Sine Waves



Computer Science

Punchline: The mean value of the product is at a maximum when you have the same frequency and the same phase:

```
In [15]: showProducts()  
Product of 5 Hz and 0 Hz
```

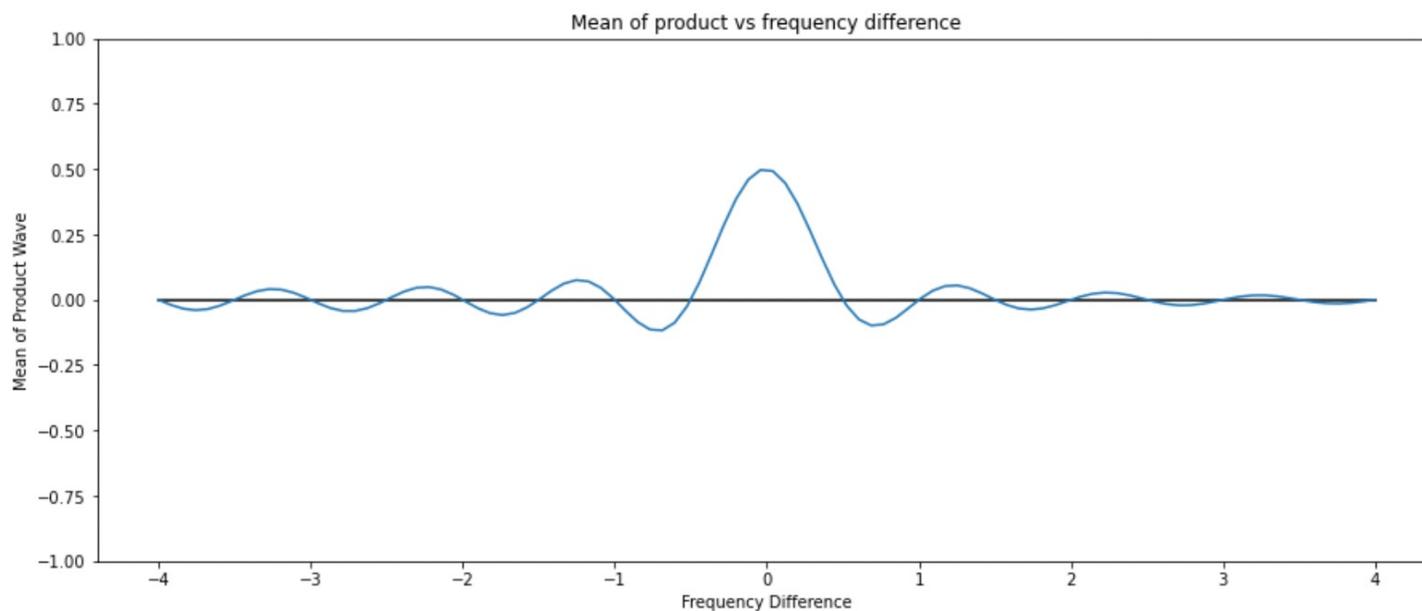


# Multiplying Sine Waves



Computer Science

And here is a graph of the mean value of the product when you change the frequency differential:



$$\text{sinc}(x) = \frac{\sin(x)}{x}$$

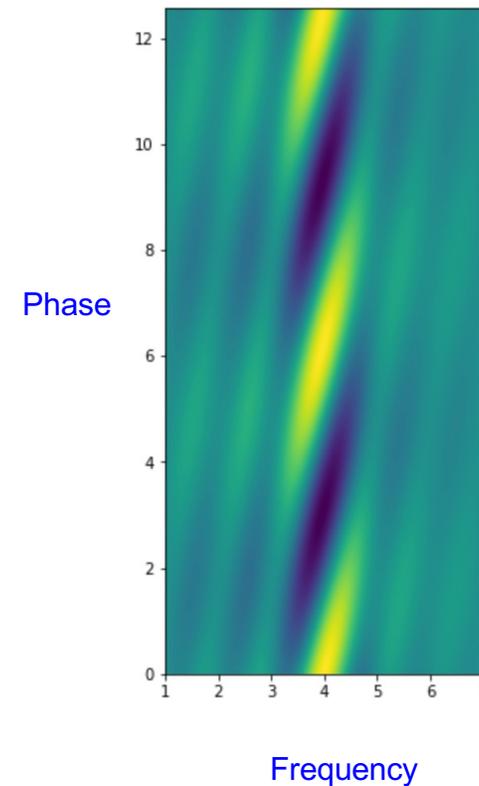
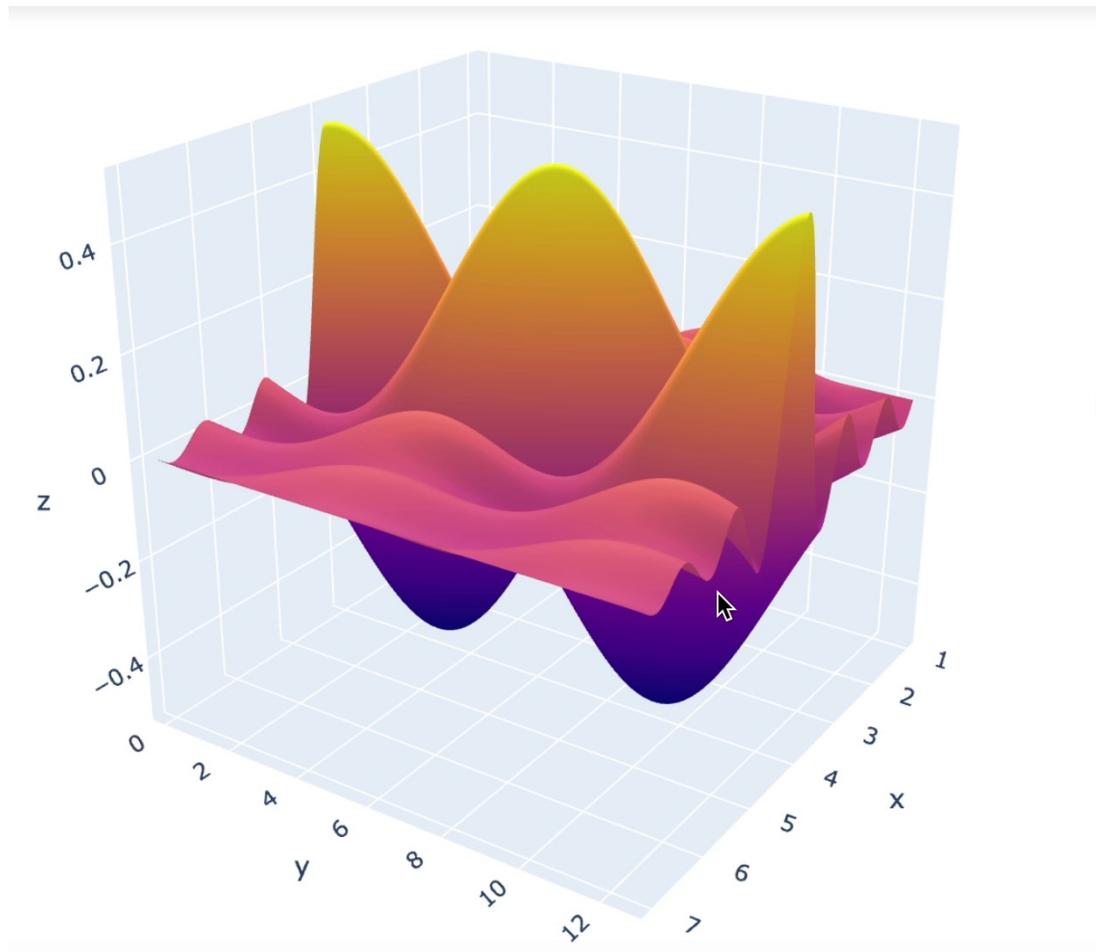
This is called a sinc wave, and it is actually a sine wave in disguise!

# Digital Audio Fundamentals: F0 determination



Computer Science

Putting this together in a 3D surface plot with  $x$  = frequency and  $y$  = phase of the “probe wave,” we can see that the maximum 0.5 is achieved only for the exact same frequency of 4 Hz and a phase difference of  $0, 2\pi, 4\pi, etc.$



# Multiplying Sine Waves

---



Computer Science

**Punchline:** To detect the phase and frequency of a sine wave, we can “probe” the wave with other sine waves of various frequencies and phases; the mean value of the product will be at a maximum of 0.5 when the frequencies and phase match.

This is one of the fundamental techniques in audio signal processing, but you’ve seen it before, in CS 237....

# Review from CS 237.....



The **Correlation Coefficient** of X and Y is defined as:

$$\rho_{X,Y} = \frac{\text{Cov}(X, Y)}{\sigma_X * \sigma_Y} = \frac{E[(X - \mu_X) * (Y - \mu_Y)]}{\sigma_X * \sigma_Y} = E\left[\frac{X - \mu_X}{\sigma_X} * \frac{Y - \mu_Y}{\sigma_Y}\right] = E[Z_X * Z_Y]$$

where  $Z_X$  and  $Z_Y$  are the standardized forms of X and Y.

To compute, it is best to use:

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$

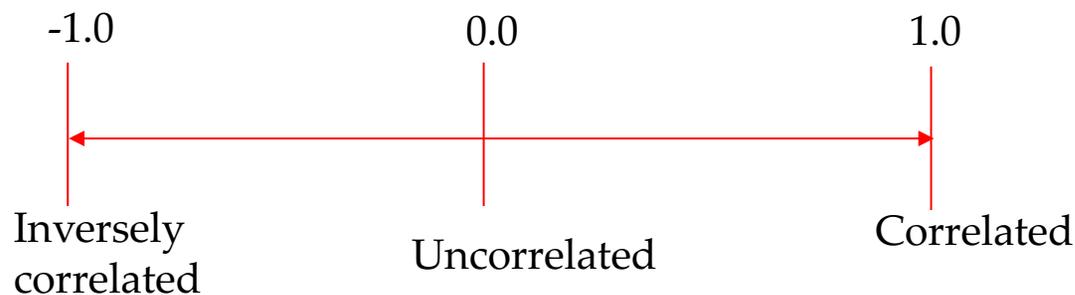
# JRVs: Covariance and Correlation Coefficient



Computer Science

If we standardize by subtracting the means and dividing by the standard deviations, the range of the **Correlation Coefficient** of X and Y is from -1 to 1:

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$



# Digital Audio Fundamentals: Correlation of Two Signals

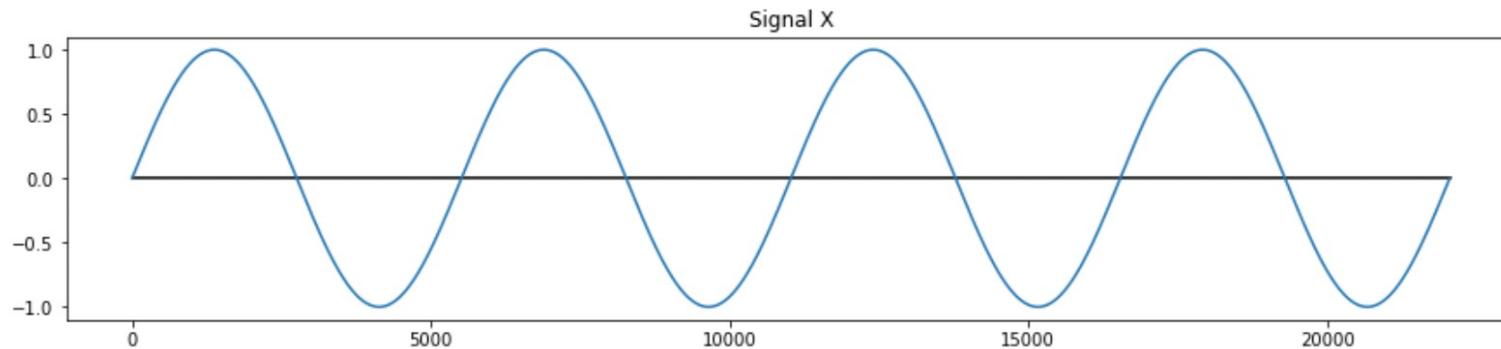


Computer Science

How to apply this to signals?

$$\rho_{X,Y} = \frac{E(X * Y) - \mu_X * \mu_Y}{\sigma_X * \sigma_Y}$$

We can simplify, since mean and standard deviation of sine waves are constants:



```
mean(X) = 0.0  
std(X) = 0.7071067811865476
```

Thus, the correlation of two sine waves is

```
: np.sqrt(1/2)  
: 0.7071067811865476
```

$$2 \cdot \text{mean}(X \cdot Y) = 2 \cdot \frac{\sum_{k=0}^N X[k] \cdot Y[k]}{N} = \frac{2}{N} \cdot X@Y$$

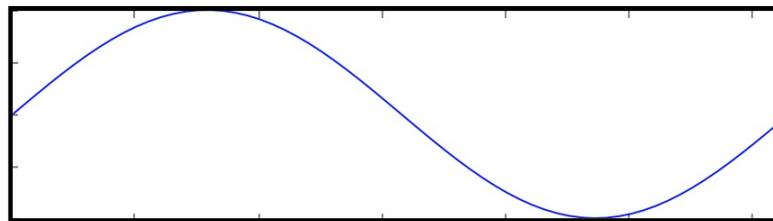
Dot Product

# Digital Audio Fundamentals: Correlation of Two Signals

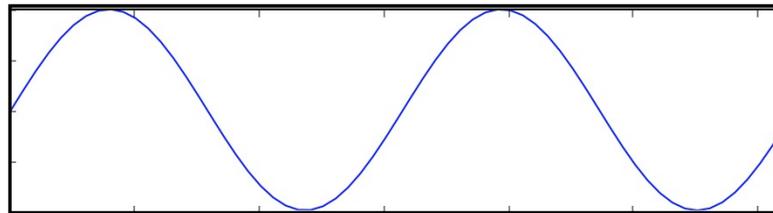


Computer Science

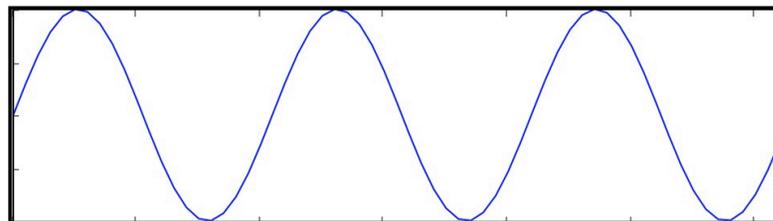
Thus, to detect the presence of sine waves in a signal, we can use "probe waves" whose frequencies are 1, 2, 3, etc. cycles in each signal window; next week we will examine the **Fourier Transform**, which is based on this idea (with complex signals).



1 Cycle



2 Cycles



3 Cycles .... Etc.

0

N-1

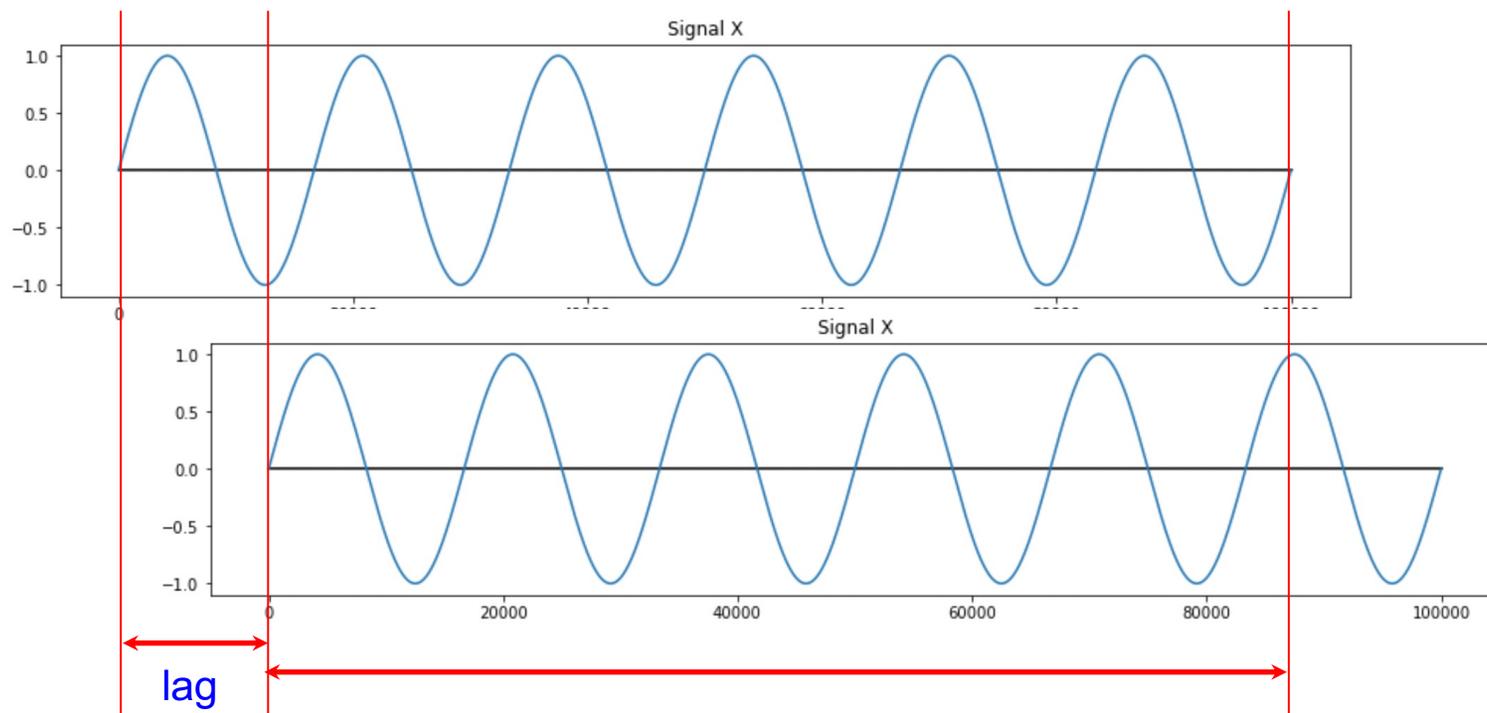
# Digital Audio Fundamentals: F0 determination



Computer Science

As a warm up, we can explore the use of correlation to find the fundamental frequency of a signal.... by using the correlation of a signal with itself...

**Auto-Correlation** is the correlation of a signal with itself, with one detail: we will shift the signal before we perform the correlation. The shift in the signal is called the “lag” and is measured in units of samples:



$$\text{AutoCorrelation}(X, \text{lag}) = \text{Correlation}( X[: -\text{lag}], X[\text{lag}:] )$$

## Digital Audio Fundamentals: F0 determination



If we plot the Auto-Correlation vs lags in the range  $0 \dots \text{len}(X)/2$ , the “peaks” in the resulting signal – remember, practically everything in this course is a signal! – show where the periods of the signal line up, and hence measure the period(s) of the periodic signal.

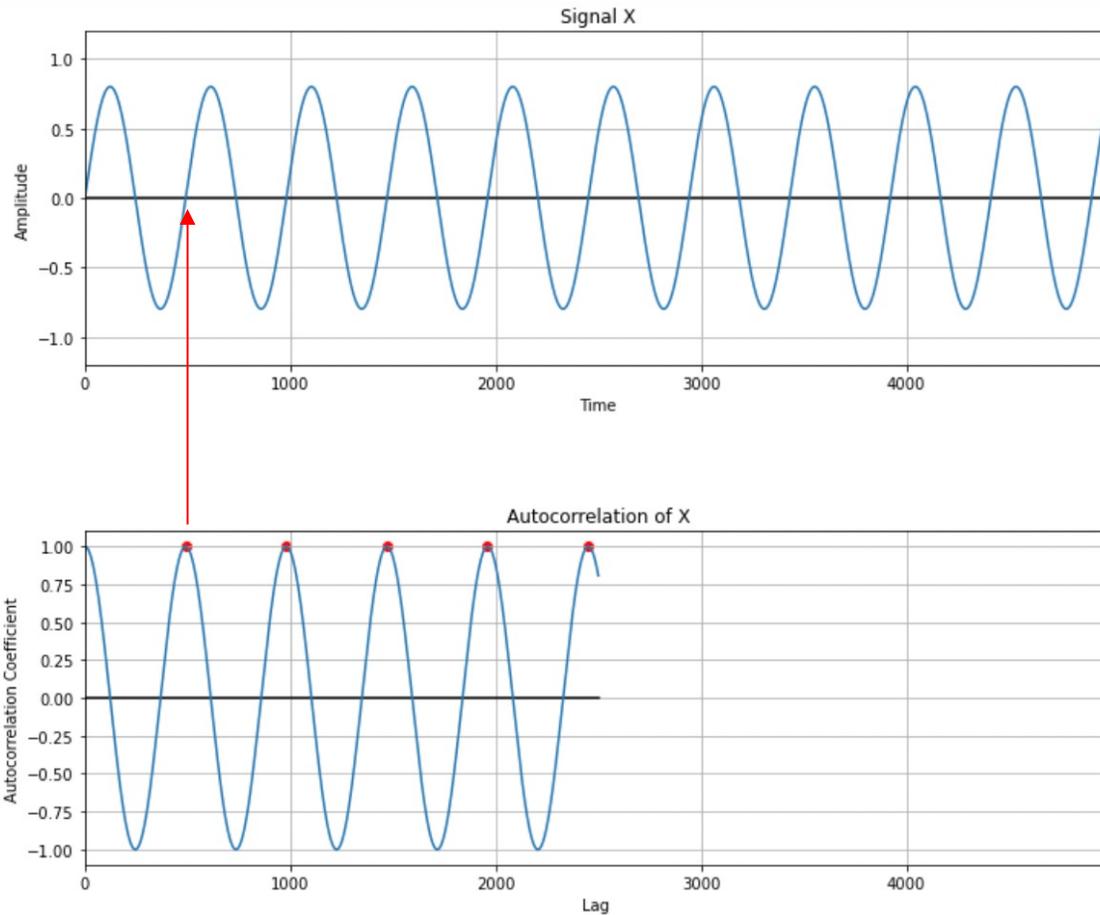
Several things to note:

- o This works (in principle) for any periodic signal (not just simple sine waves);
- o It is only practical to measure signals which have at least two full periods, which means you only consider lags up to  $\text{len}(X)/2$ ;
- o The “fundamental frequency”  $f_0$  is the inverse of the shortest period; you will typically get peaks at  $f_0, 2f_0, 3f_0, \dots$

# Digital Audio Fundamentals: F0 determination



## Examples of Peak Picking to determine F0:



Period found at 490 time units.  
Fundamental Frequency = 90.0 Hz

## Calculation of F0:

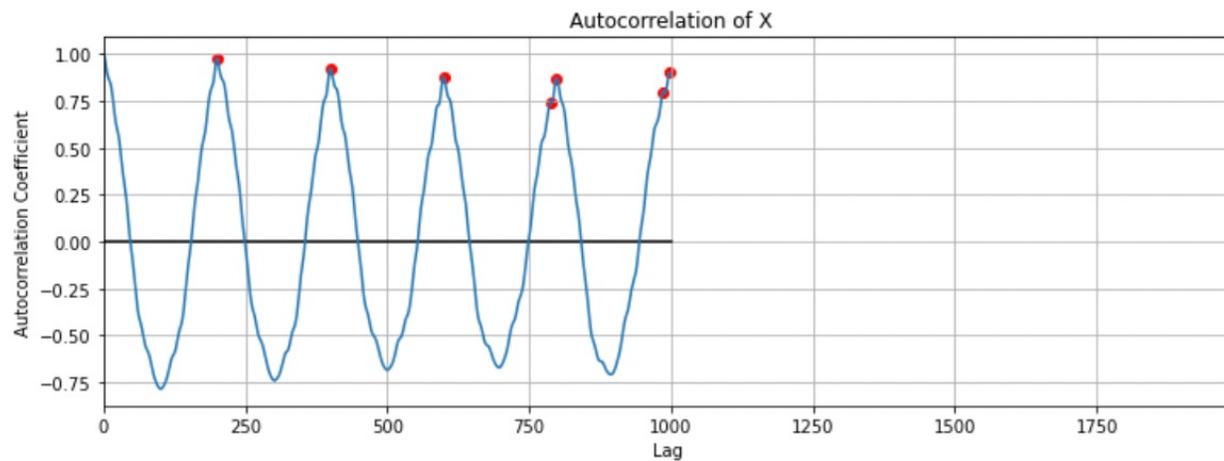
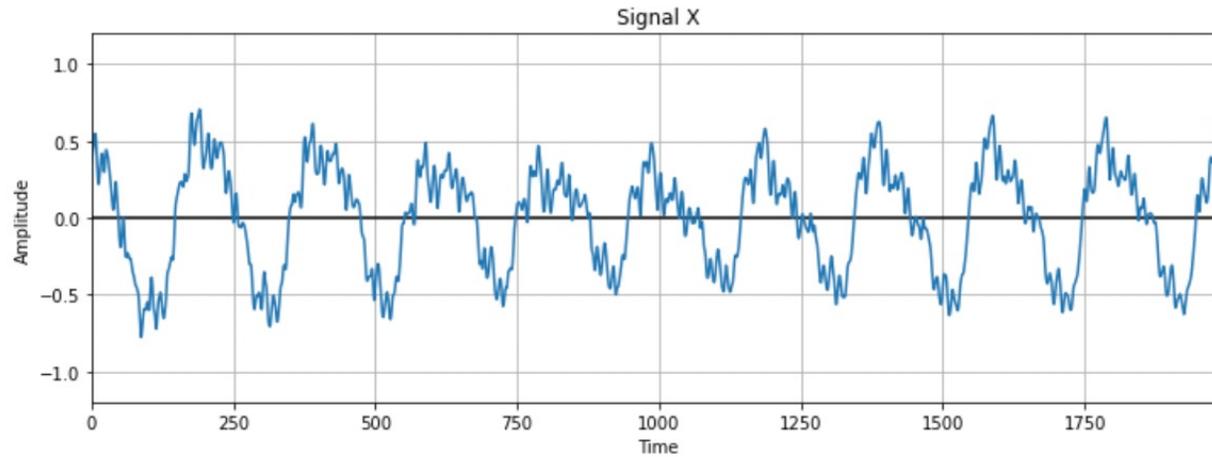
$$\text{Period in Seconds} = \frac{\text{Number of Samples}}{\text{Sample Rate}}$$

$$\text{Frequency in Hz} = \frac{1}{\text{Period in Seconds}}$$

# Digital Audio Fundamentals: F0 determination



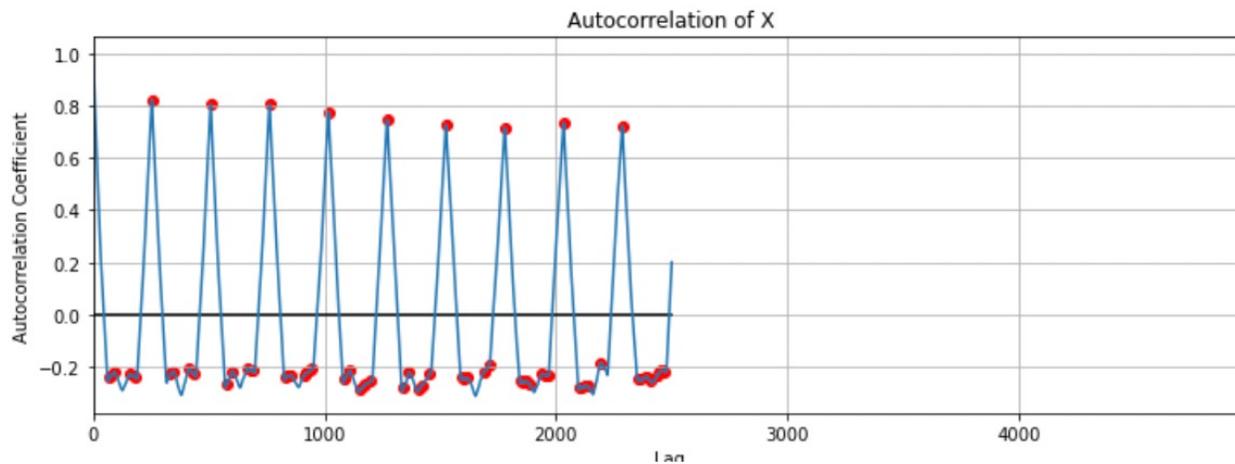
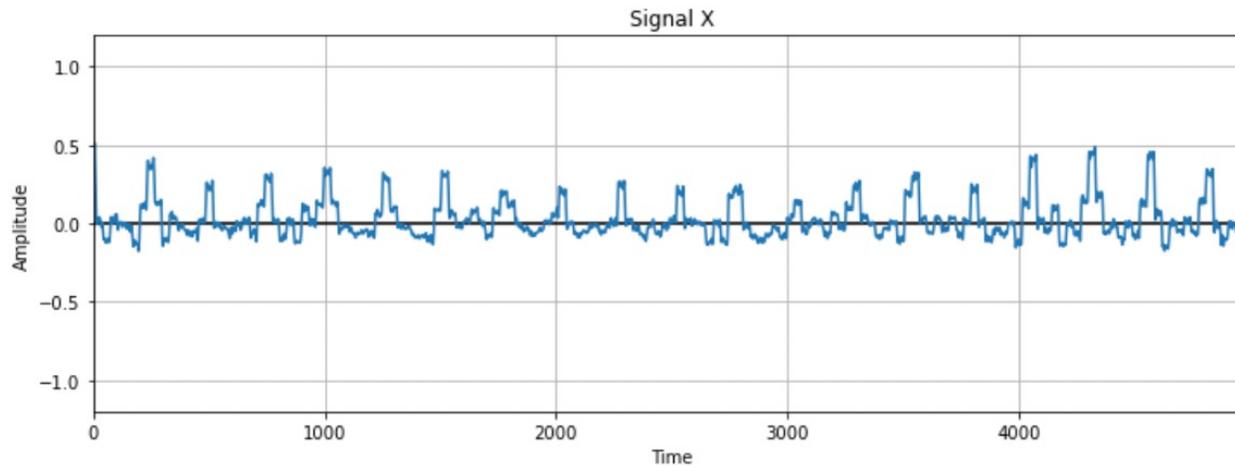
## Examples of Peak Picking to determine F0:



# Digital Audio Fundamentals: F0 determination



## Examples of Peak Picking to determine F0:



# Digital Audio Fundamentals: F0 determination



## Examples of Peak Picking to determine F0:

