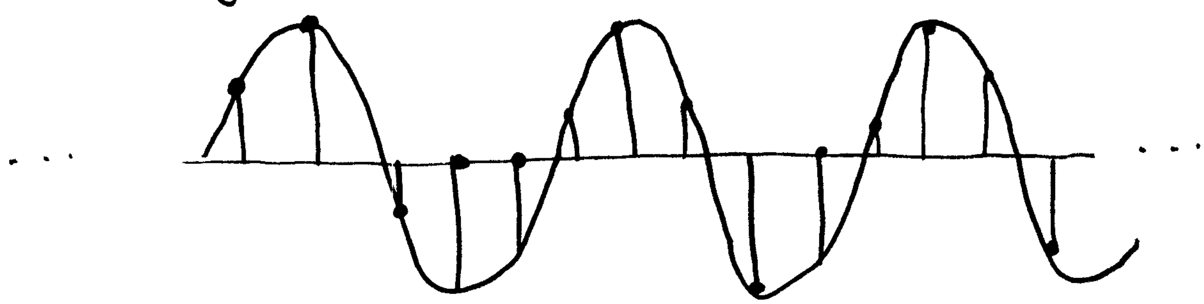
FFT & Signal Processing (1 & 2 - dim.)

Closer look at
- sampling/discretization 1/2-dim.
- Fourier representation/ Frequency domain

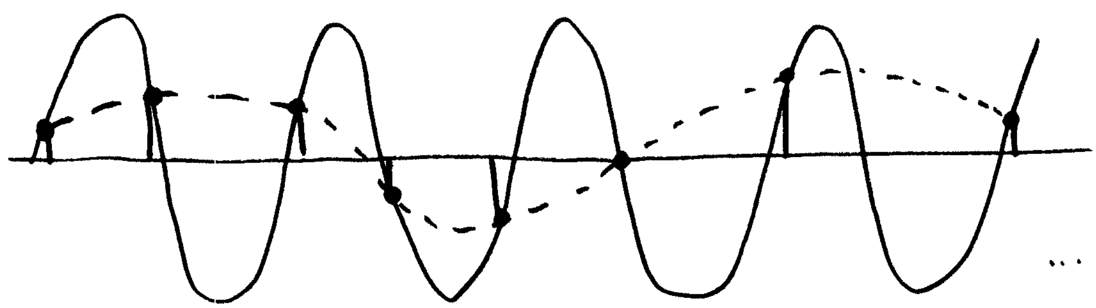
Sampling: We've used grids for all the differential eqn. work. think about continuous fctns. as sums of sinusoids, and sample:

sampling relatively fast



... OK

Sampling too slowly

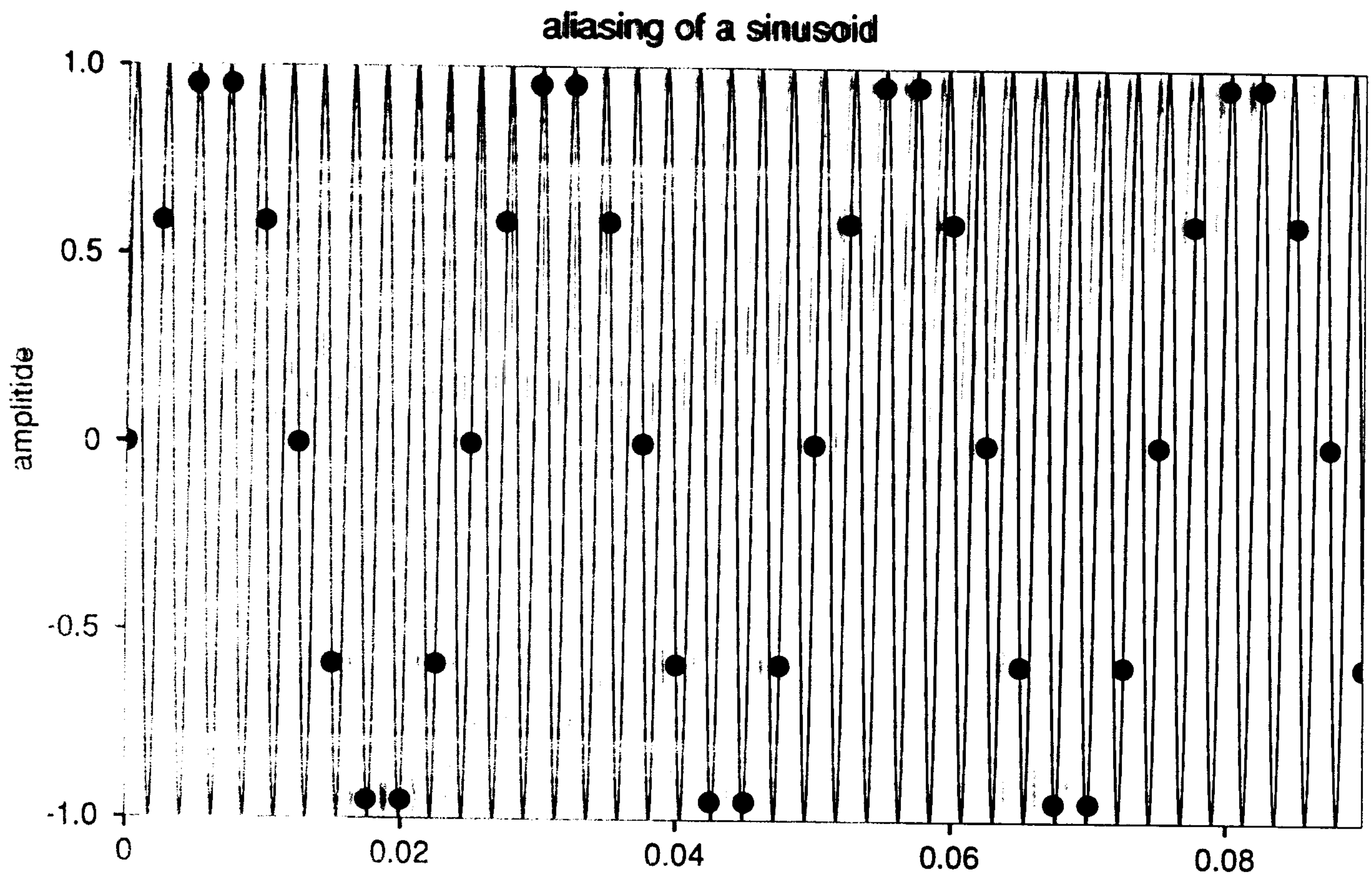


... NOT OK

We are deceived into thinking this a lower frequency.

A high frequency is masquerading as a lower one.

We say the lower frequency is an alias of the higher.


aliasing of a sinusoid
amplitide
1.0
0.5
0
-0.5
-1.0
0
0.02
0.04
0.06
0.08

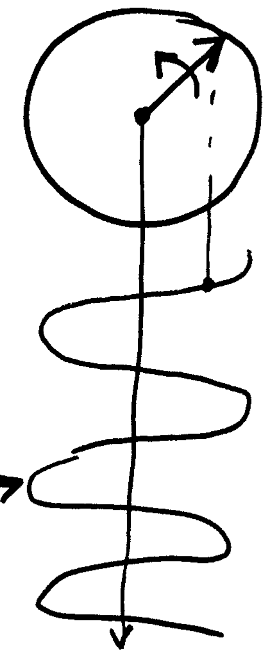
As usual, it's more illuminating to view sinusoids as projections of points moving around circles.

Phasors, complex exponential representation

→ We used this in Von Neumann's stability analysis
→ We used this to solve wave eqn. for string in separation of variables

$e^{j\omega t}$

can be time or space



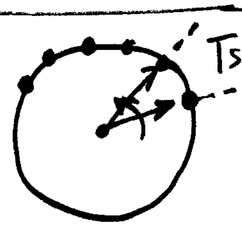
Projection is $\cos \omega t$

ω radians/sec

f Hz = cycles/sec
$= \omega / 2\pi$

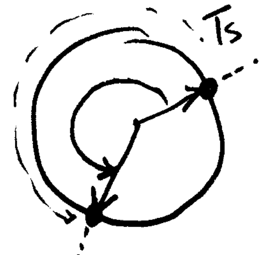
T = period $= \frac{1}{f} = \frac{2\pi}{\omega}$ sec/cycle

Sampling fast enough



T_s = sampling interval
$= 1/f_s = 2\pi/\omega_s$

not sampling fast enough



appears to going backward!

Condition for unambiguous resolution of frequency is

"signal" freq.

$T_s \le T/2$

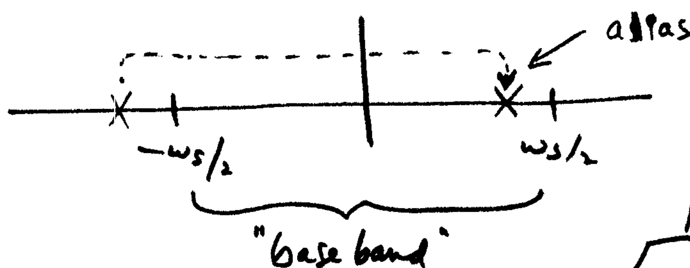
$\boxed{f_s \ge 2 \cdot f}$

Nyquist's criterion: Must sample at a rate at least twice the highest frequency in signal.

Put another way, if we sample at rate ω_s rad/sec, the highest allowed signal frequency is

$$\omega_s/2 = \text{"Nyquist frequency"}$$

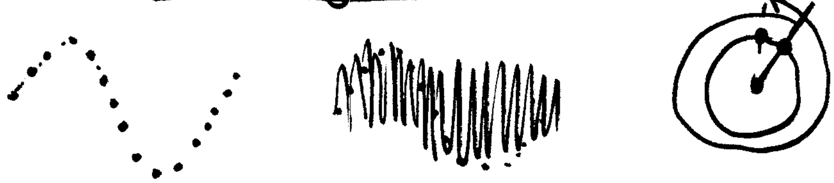
When sampling at freq. ω_s, we therefore can consider all frequencies as lying between $-\omega_s/2$ and $+\omega_s/2$:



pos. freq. part   neg. freq. part

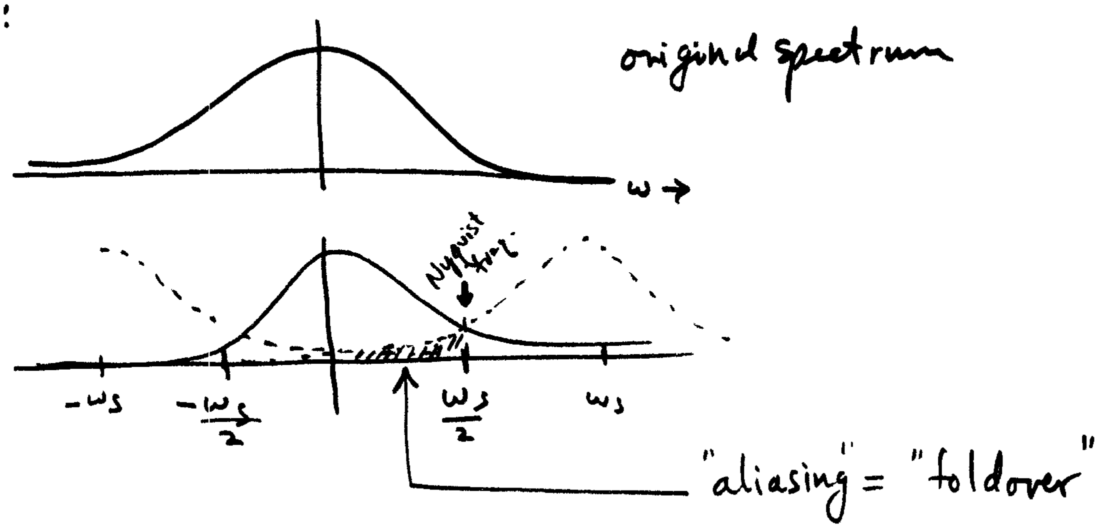
negative frequencies?? $\rightarrow$ $\cos\omega t = \frac{1}{2}e^{j\omega t} + \frac{1}{2}e^{-j\omega t}$

Yet another way to look at aliasing: All frequencies that differ by an integer multiple of ω_s radians/sec are indistinguishable.



more than one rotation per sample

Aliasing:



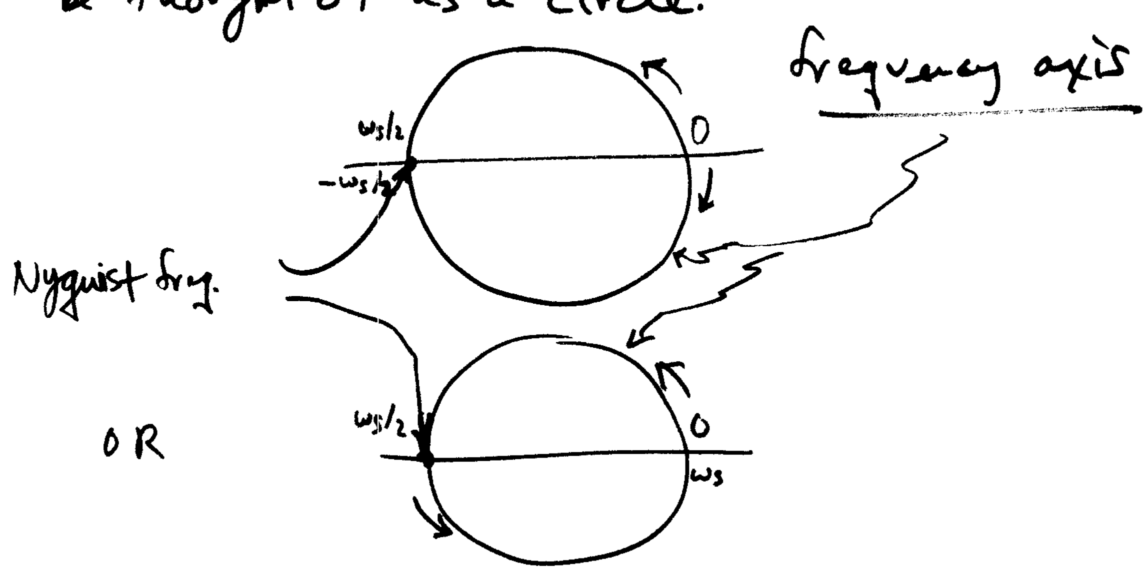
<u>Audio</u> aliasing is very disturbing because harmonic components get aliased to <u>non-harmon</u>ic components

dealt with by <u>pre-filtering</u>, removing components above $\omega_s/2$ <u>before</u> sampling

<u>Images</u> aliasing often shows up as disturbing herringbone patterns (<u>Moire</u> patterns).

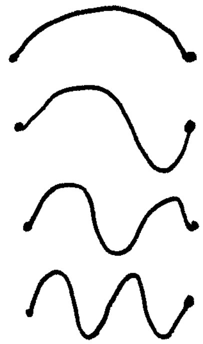
For example, striped shirt on TV } raster scan

<u>Note</u>: the frequency axis after sampling can therefore also be thought of as a circle:



Fourier Analysis

Recall vibrating string: modes

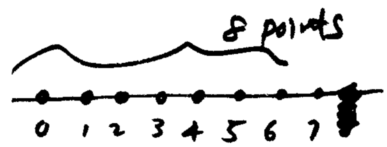


... etc.

the shape at any time is a linear combination of these sinusoids. this is a general principle - any waveform can be represented as a sum of sinusoids.

Sampled version is called the Discrete Fourier Transform.

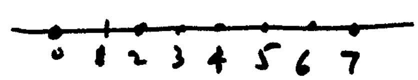
finite-extent sampled signal



$t = 0, \dots, N-1$

$x_t, \; t = 0, \dots, N-1$

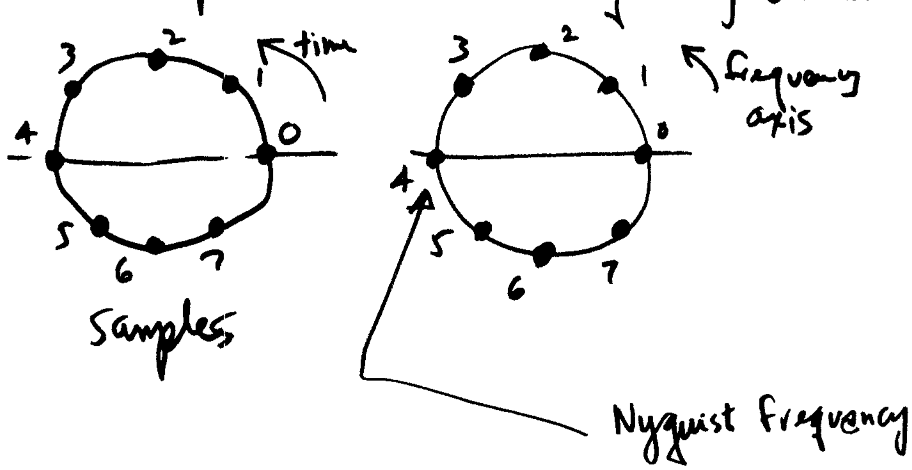
set of frequencies used for representation



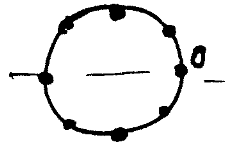
$k = 0, \dots, N-1$

frequencies $\frac{k 2\pi}{N}$, $k = 0, \dots, N-1$

think of samples' domain and frequency domain on circles



Algebraically:



"signal"
x_t, $t=0,\dots,N-1$

frequencies
$e^{jk2\pi/N}$, $k=0,\dots,N-1$

Fourier representation

$$\boxed{x_t = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j(k2\pi/N)t}}$$

$\underline{\text{amount}}$ of frequency $k\frac{2\pi}{N}$
= spectral component
= Fourier component

How to find Fourier components:

N eqns & N unknowns → uniquely determined.

[If you know linear algebra, the frequency components $e^{jk(2\pi/N)t}$ form an ~~orthonormal~~ orthogonal basis]

turns out that

$$\boxed{X_k = \sum_{t=0}^{N-1} x_t e^{-j(k2\pi/N)t}}$$

Forward DFT

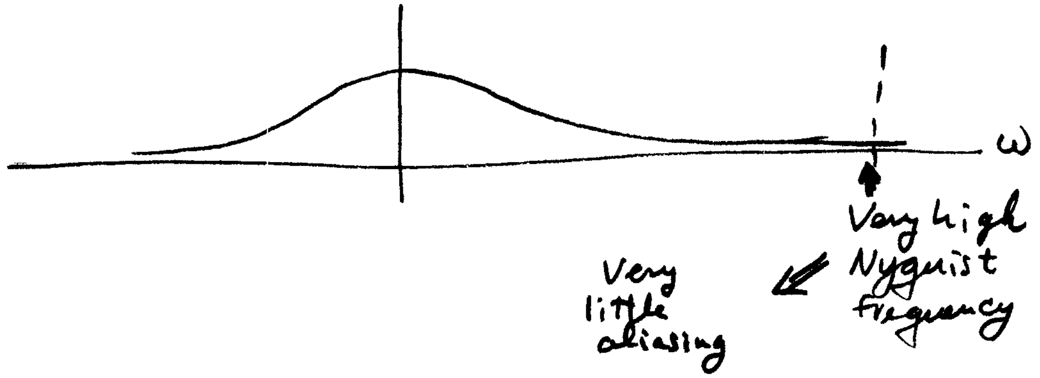
$$\boxed{x_t = \frac{1}{N}\sum_{k=0}^{N-1} X_k e^{j(k2\pi/N)t}}$$

Inverse DFT

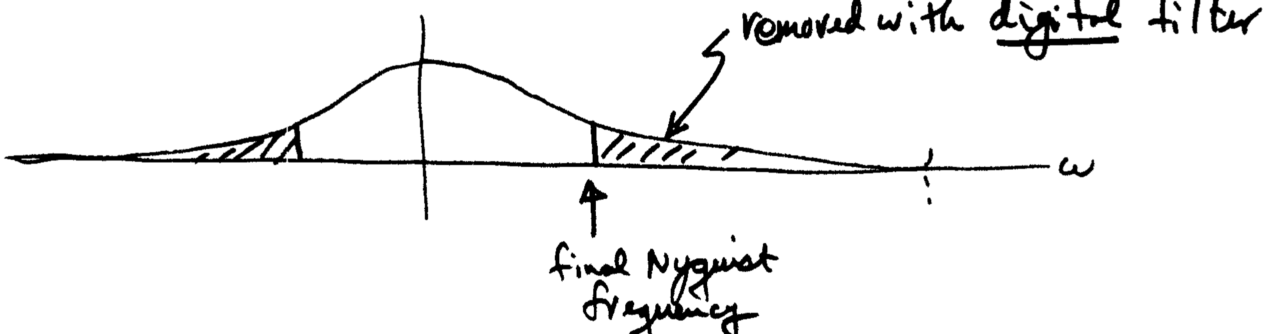
the $\frac{1}{N}$ shows up one place or another; conventionally we put it in the inverse DFT

typically, we need 1024 or 2048 to get a good frequency representation.

Oversampling: Idea: 1) sample much faster than necessary



2) then filter out components above the final Nyquist frequency, using cheap digital filtering

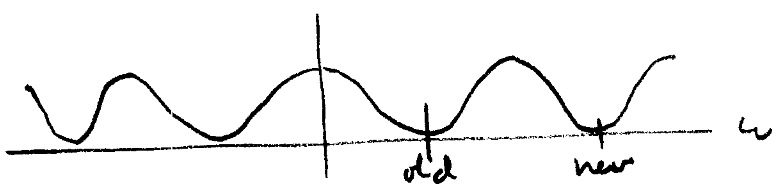


3) then reduce sampling rate (sub-sample) to final, practical rate.

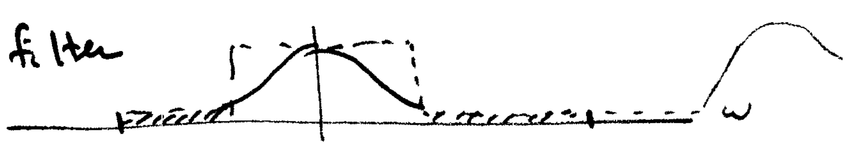
---

that's analog-to-digital conversion.
A similar idea is used in CD-players on digital-to-analog conversion:

1) Increase the Nyquist rate in the digital domain (by inserting zeros)



2) Digital filter



3) convert at higher rate ⇒ much less aliasing

Naïve algorithm N points @ N multiplication per point

Divide & Conquer (like merge sort)

1) Divide sequence into two parts
2) FFT each subsequence
3) merge in linear time

time for N-pt. transform

$$= T(N) = \underbrace{2T(N/2)}_{\text{recursive calls to half-sized problems}} + \underbrace{cN}_{\text{merge time}}$$

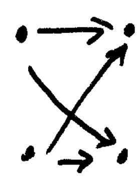
Telescope

$$T(N) = cN + \underbrace{2\left[c\frac{N}{2} + 2\left[\cancel{\ }c\frac{N}{4} + 2\left[c\frac{N}{8} + \cdots\right.\right.\right.}_{\log_2 N \text{ stages}}$$

$\Rightarrow$

$$\boxed{T(N) = O(N \log N)}$$

Decimation-in-time algorithm divides sequence into even- and odd-numbered subsequences. Requires "shuffle" to reassemble.

Merge step look like  repeated down lists

"butterfly"