Perception & Multimedia Computing

Week 12 - Fourier analysis

Michael Zbyszyński
Lecturer, Department of Computing
Goldsmiths University of London

(From last week) Using frequency content to reason about audio perception

Rule of Thumb #1

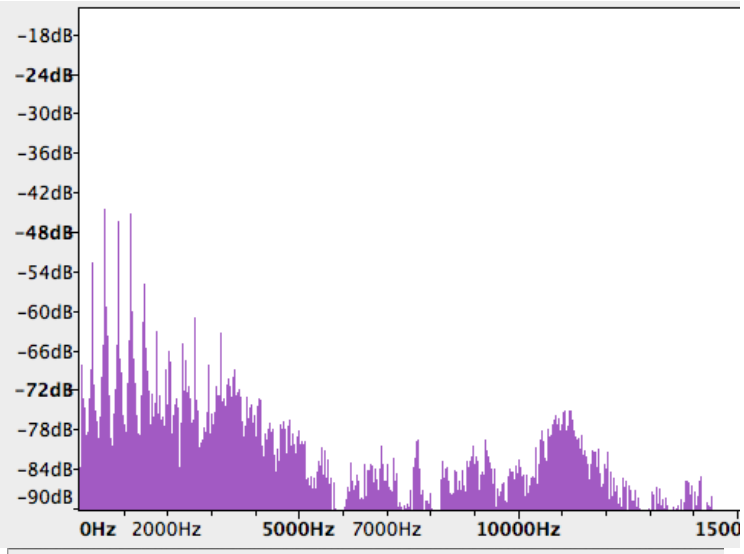
Pitched sounds have sinusoidal components that are harmonically related.

This is due to the physics of strings and air columns. When bowed / plucked / blown / etc., they will vibrate in certain way and not others.

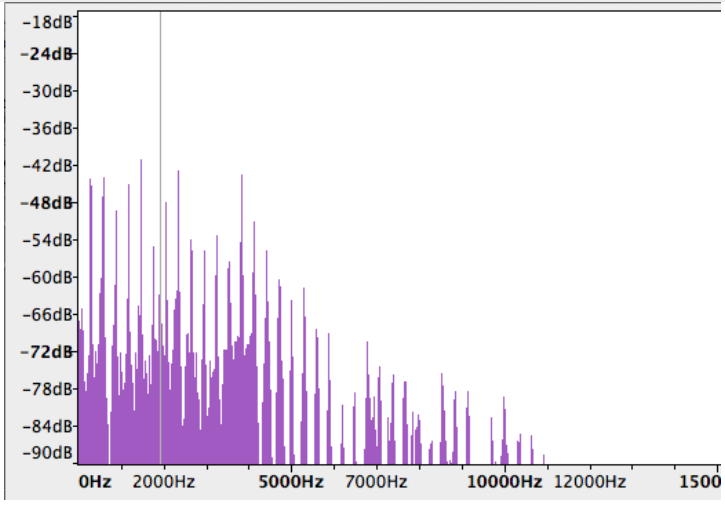
Rule of Thumb #2

The pitch we hear is determined by the fundamental frequency

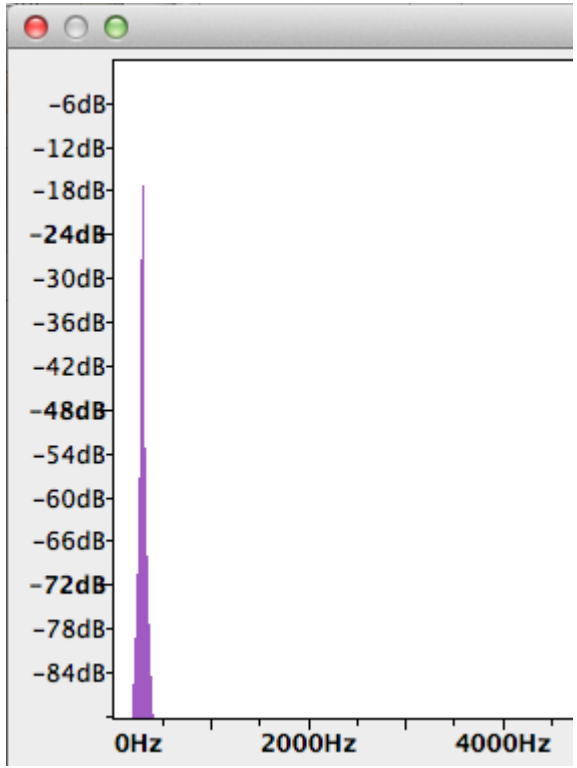
Flute playing “D” above middle C



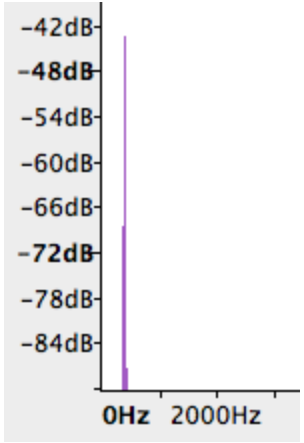
Violin playing same note



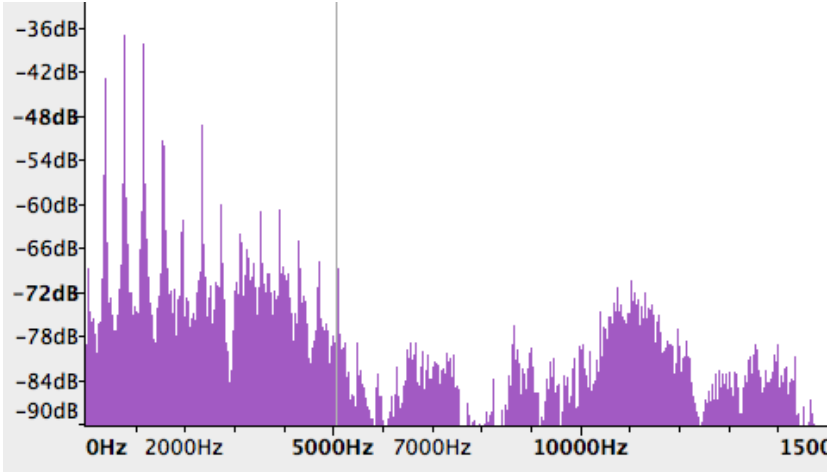
Sine wave at 294 Hz



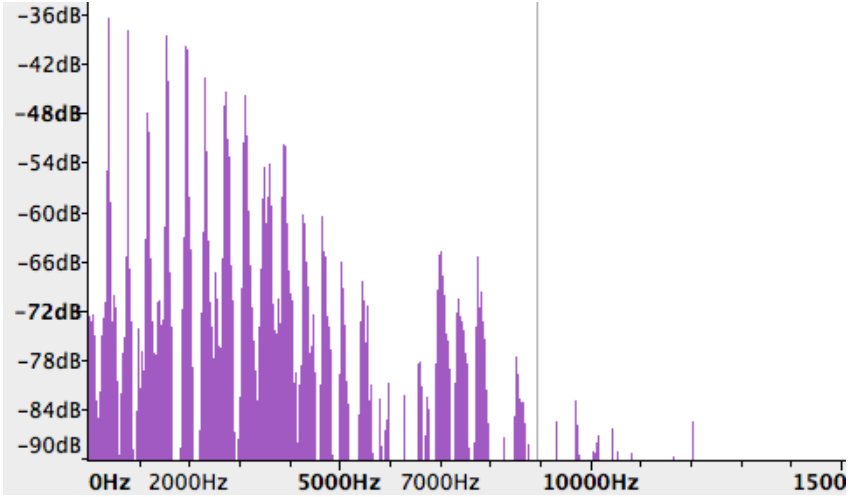
Sine
at 392 Hz



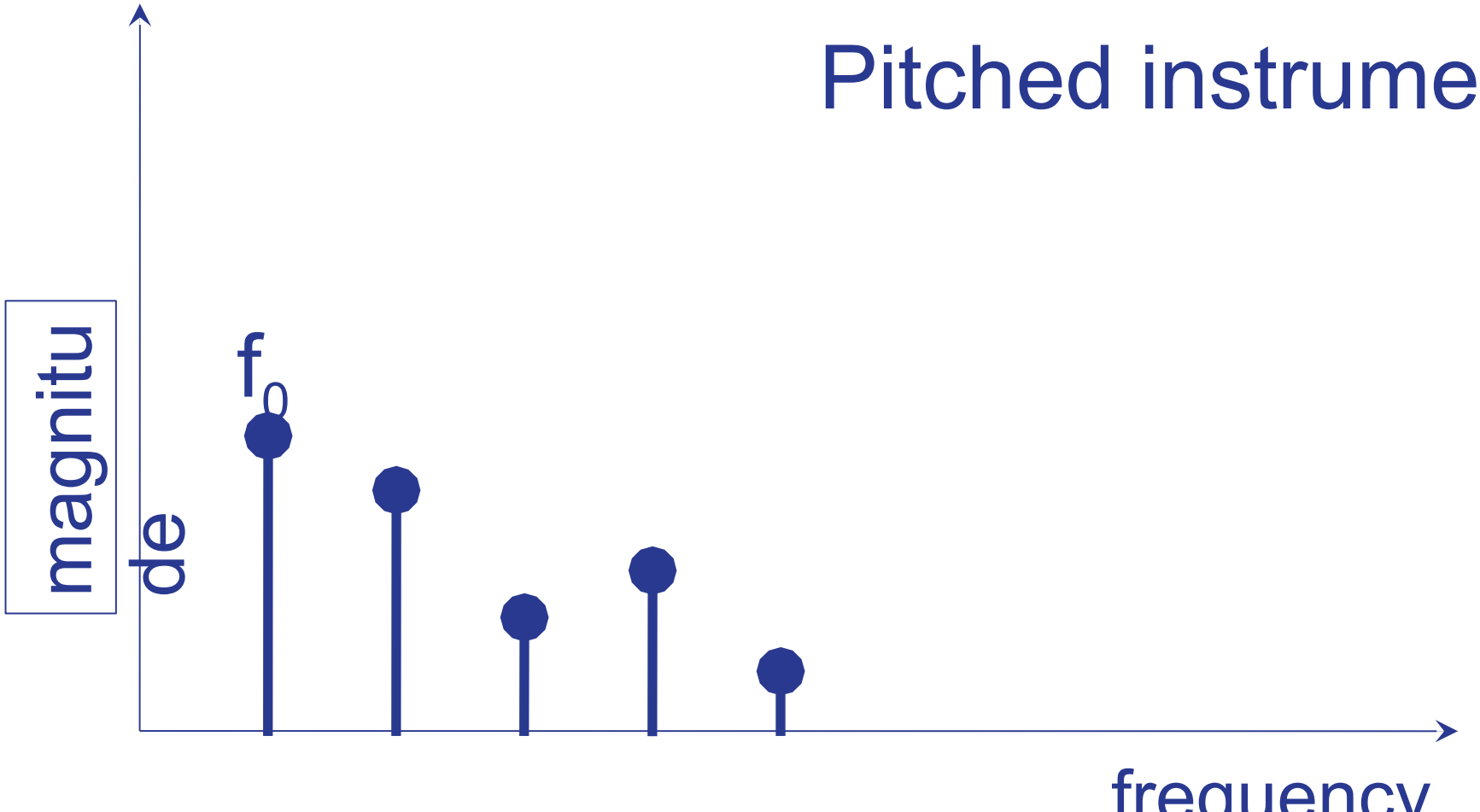
Flute



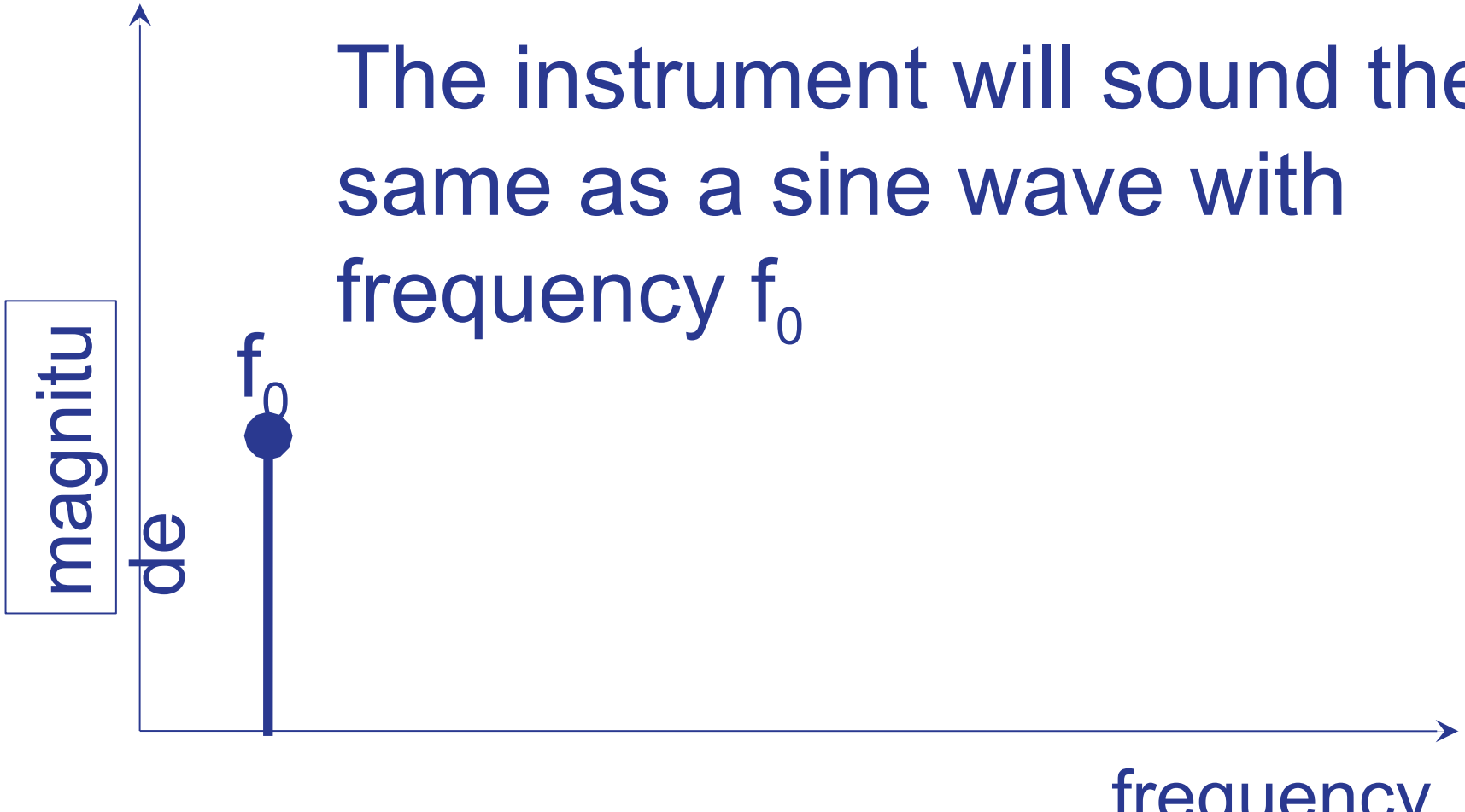
Violin



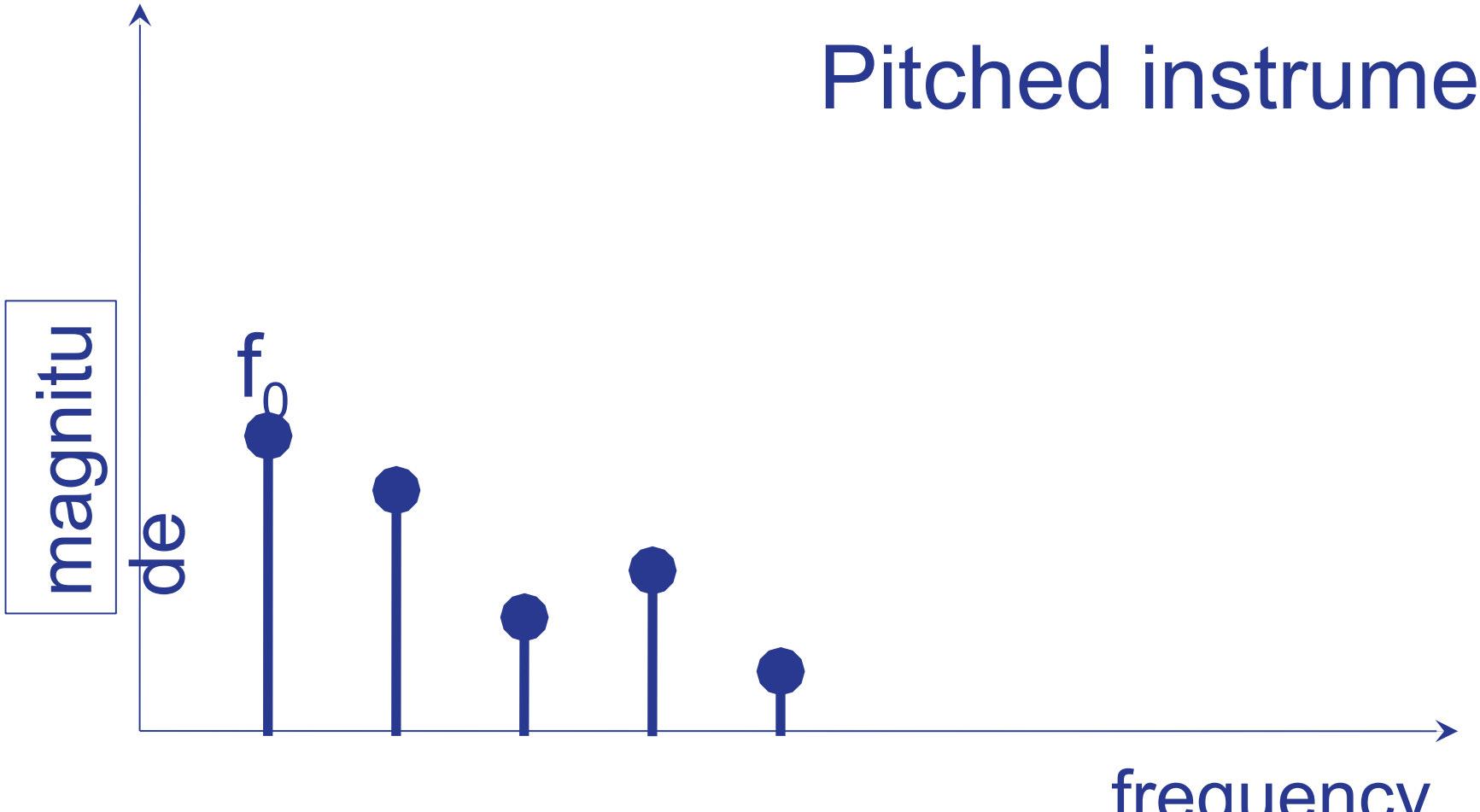
Pitched instrument



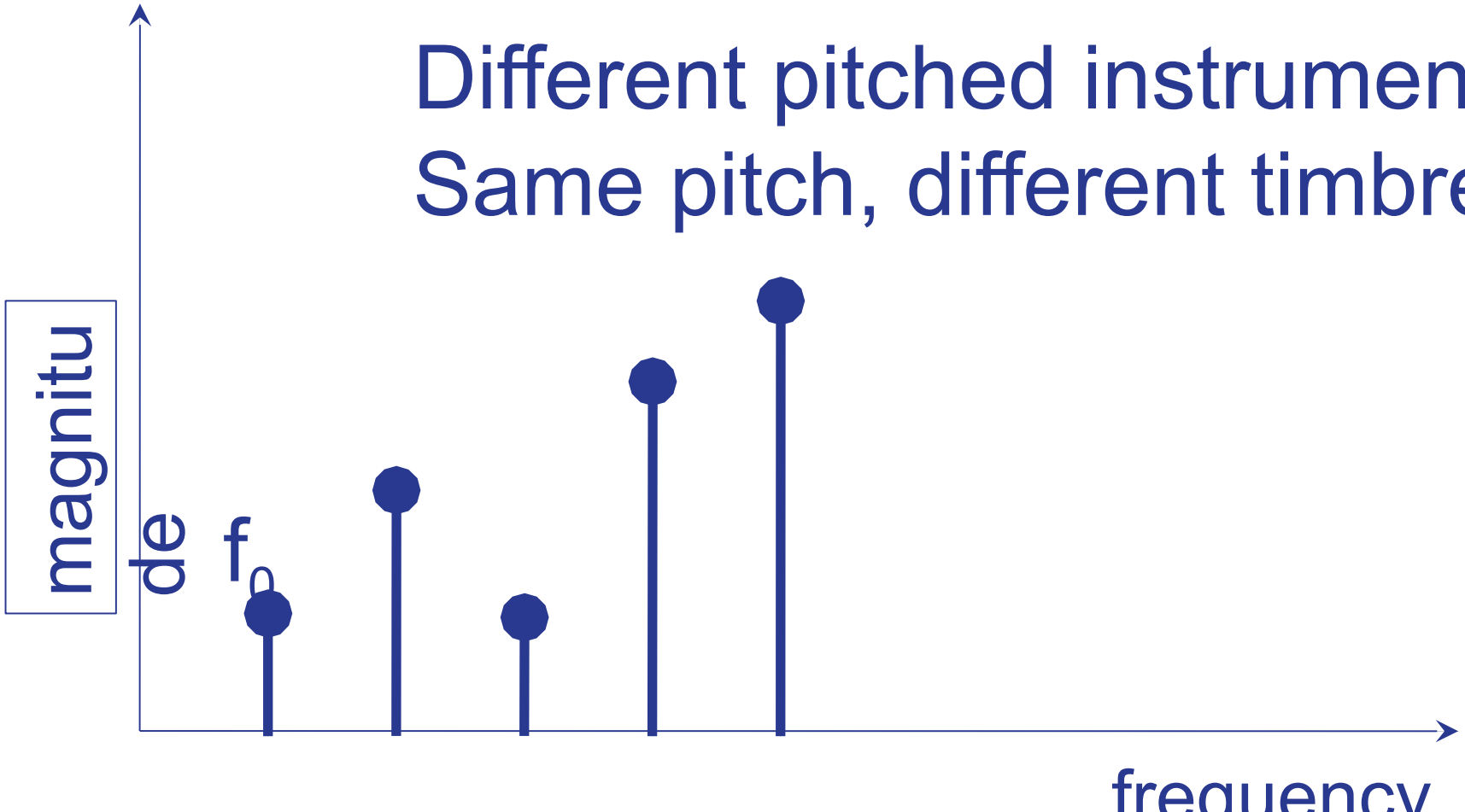
The instrument will sound the same as a sine wave with frequency f_0

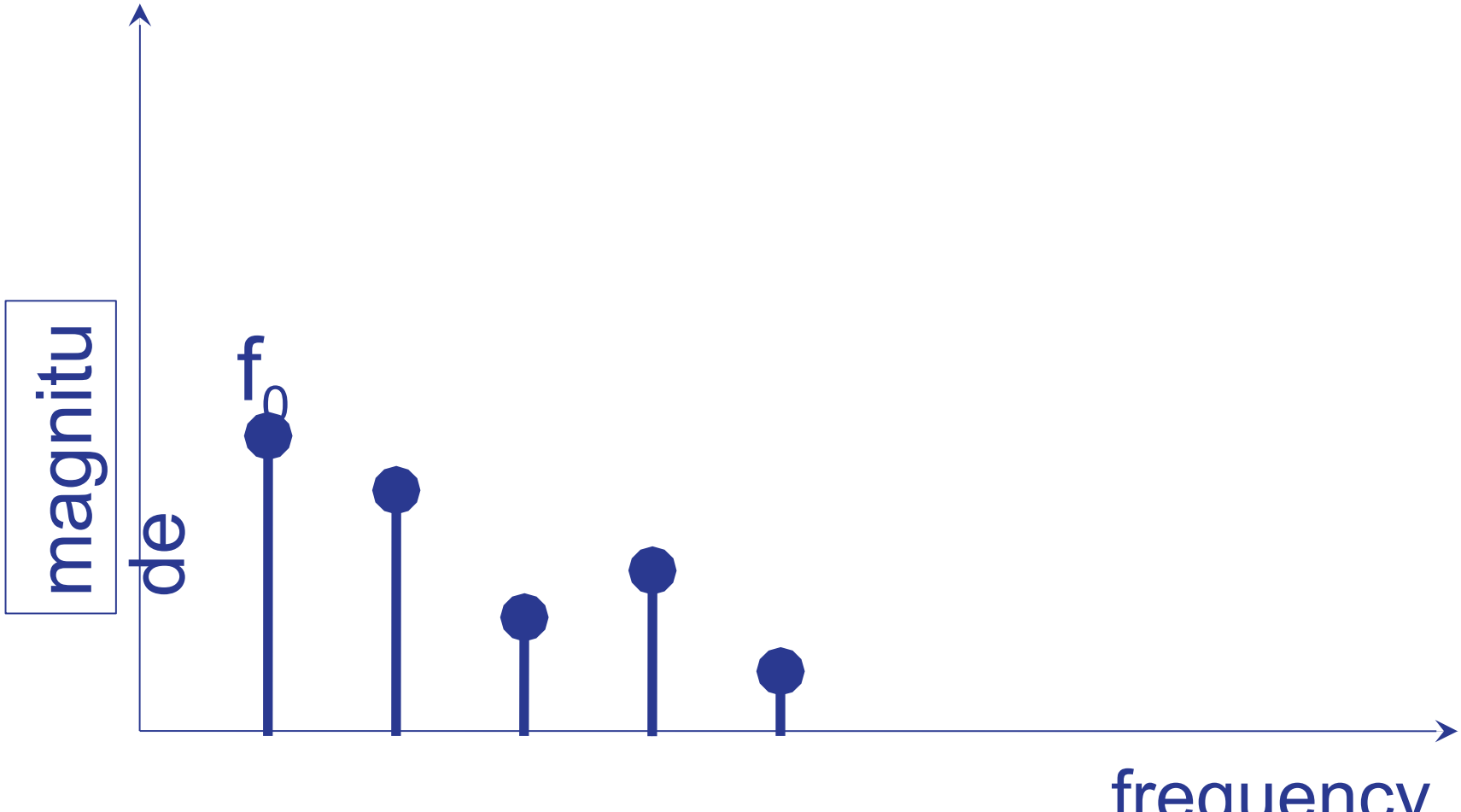


Pitched instrument

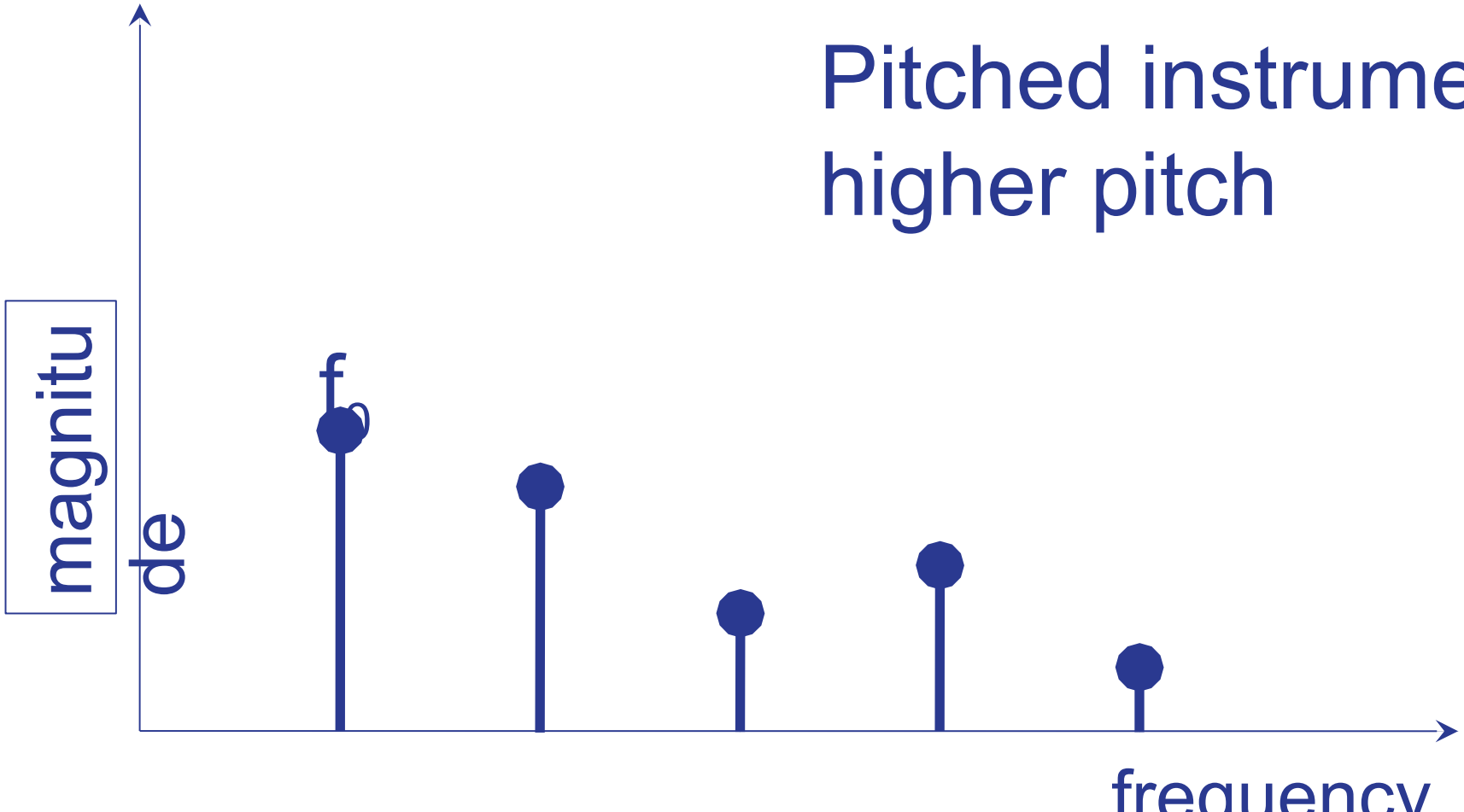


Different pitched instrument
Same pitch, different timbre




magnitu
de
f0
frequency

Pitched instrument
higher pitch

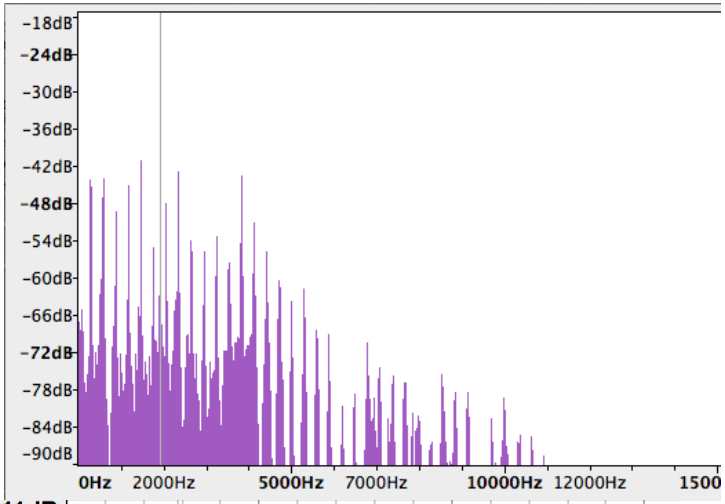


Rule of Thumb #3

The degree to which a sound's frequencies are harmonically related influences the degree to which we hear it as "pitched."

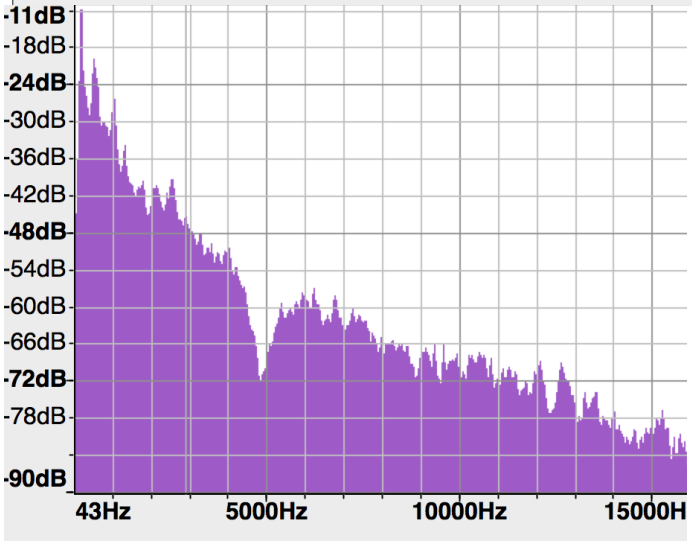
(less harmonically related = less pitched)

Violin



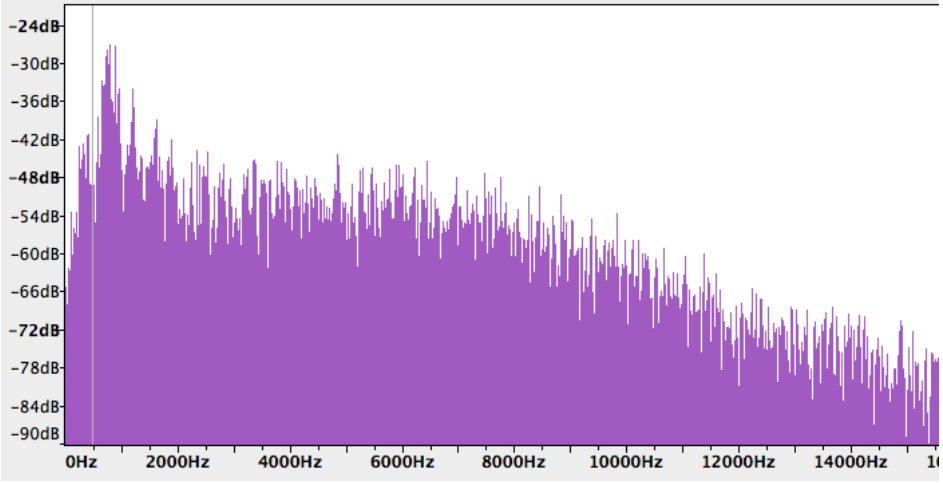
Peaks at approx.
294Hz, 589Hz, 884, ..

Ciblon

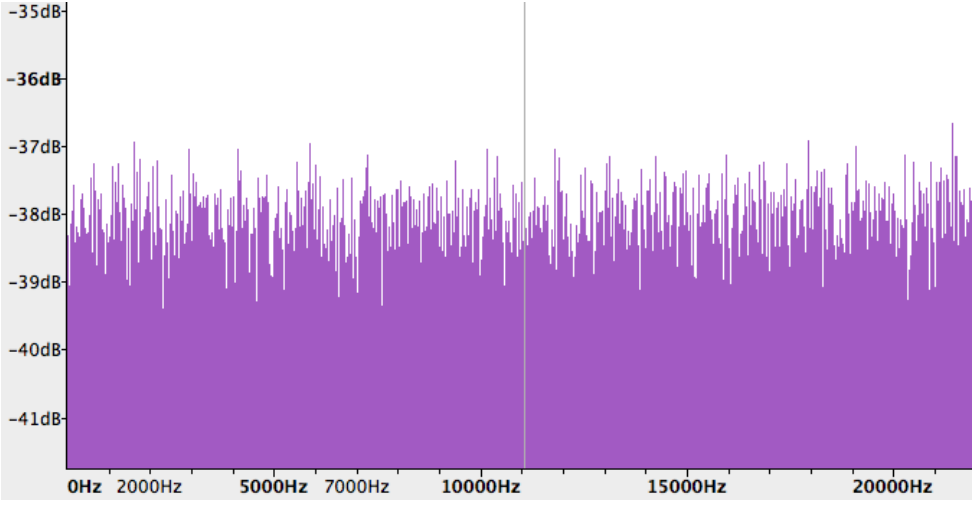


Peaks at approx.
195Hz, 522Hz,
1320Hz, 2560Hz…

Snare

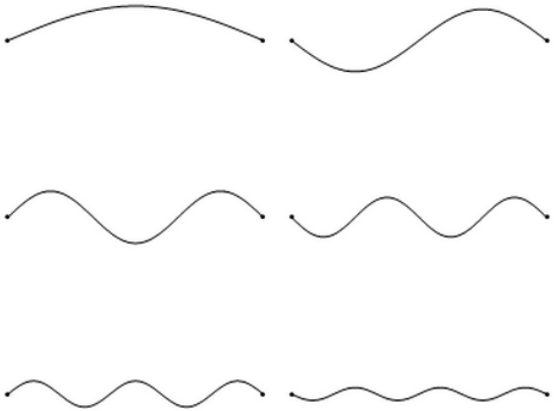


White noise



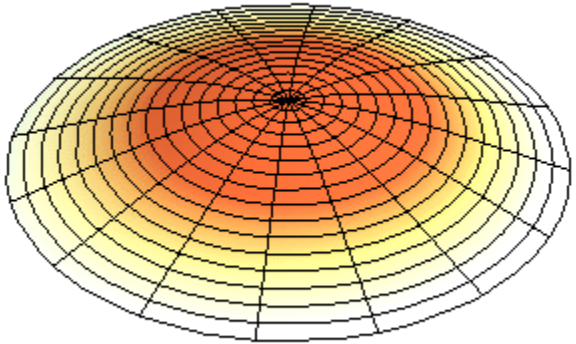
Why aren’t drums pitched?

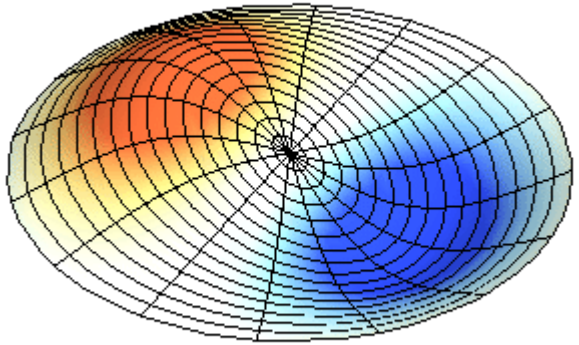
- Strings, air columns vibrate at harmonics:



- 2D surfaces do not (*inharmonicity*)

Modes on a drum head



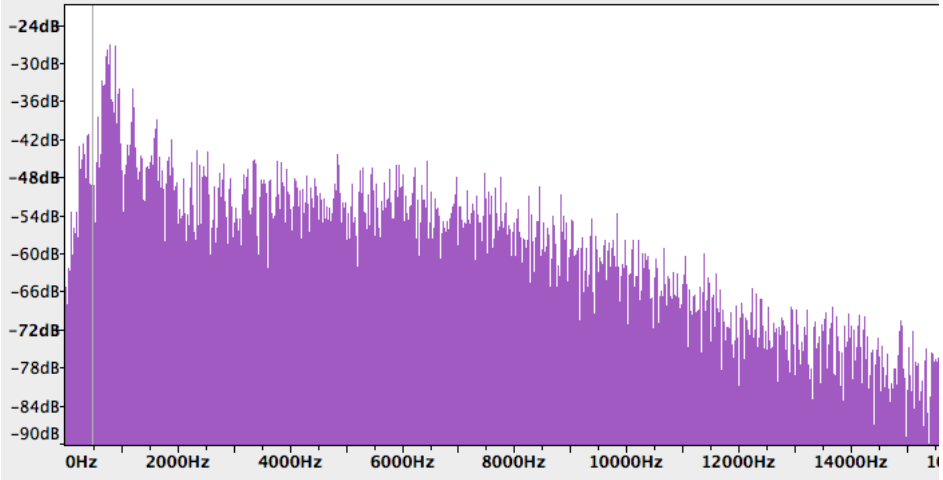


(and many more)

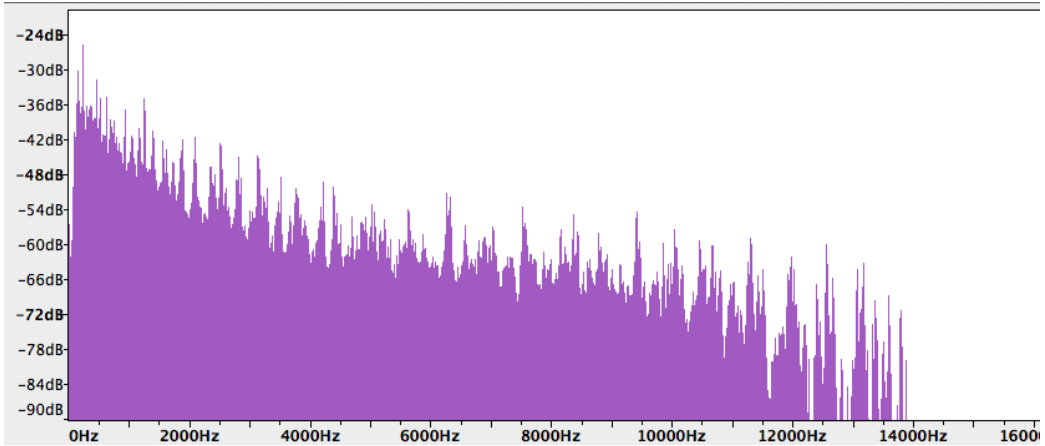
http://www.acs.psu.edu/drussell/demos/membranecircle/circle.html

One sound or many?

Snare



Orchestra



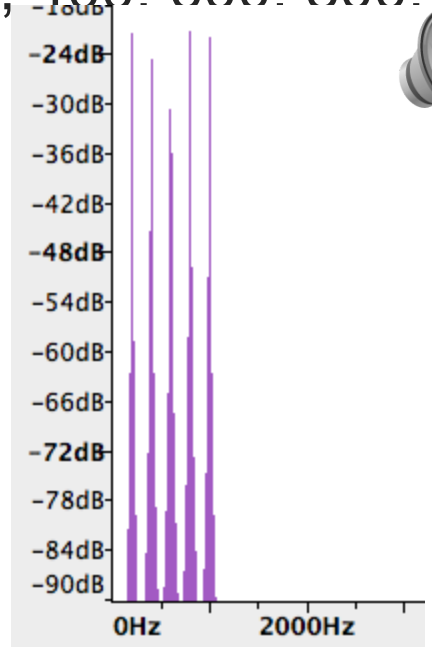
Single or multiple sound sources?

These make it more likely to hear a single sound ->

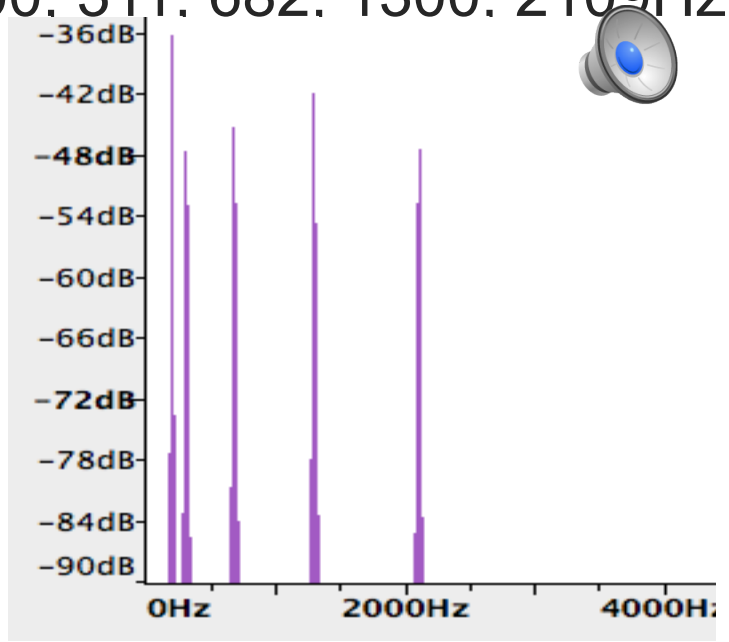
- Harmonic relationship
- Shared onset time
- Shared location
- Shared changes in amplitude (envelope)
- Shared changes in frequency (vibrato)

Harmonic relationship

Sound with 5 harmonically-related partials (200, 400, 600, 800, 1000Hz)



Sound with 5 inharmonically-related partials (200, 311, 682, 1300, 2109Hz)



Onset time (when does the sound begin?)

8 harmonics, shared onset time

Same 8 harmonics, different onset times

Shared location

8 Harmonics, 4 panned left and 4 panned right

4 moving left→ right, other 4 moving right → left

Same 4, all centre

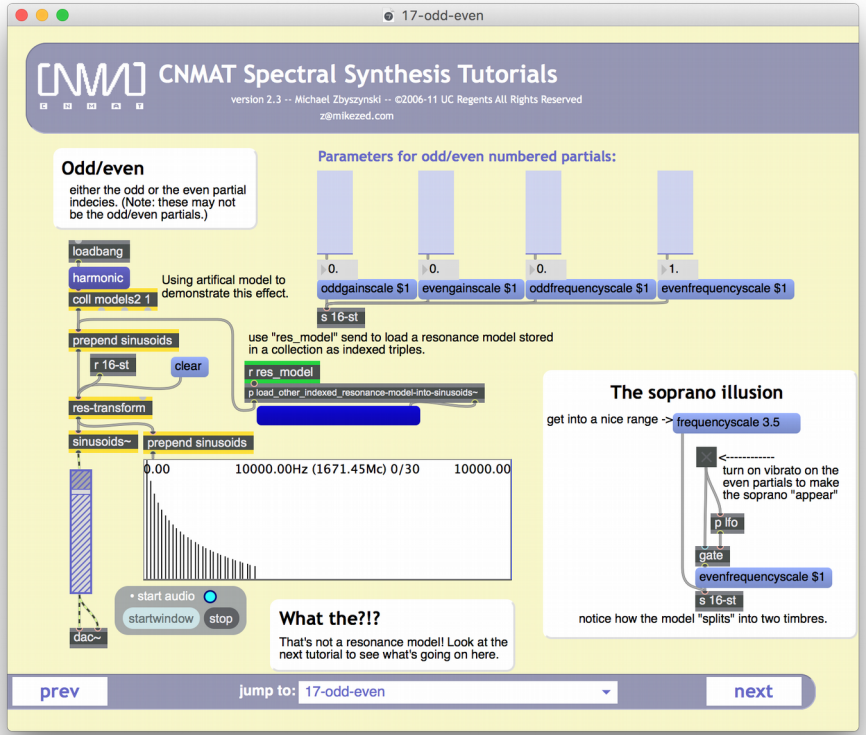
Shared changes in amplitude

Envelope:

Describes changes in the overall amplitude of a signal over time:

Envelope demos

Shared changes in frequency: vibrato demos



Vibrato demo: the soprano illusion

Single or multiple sound sources?

The following make it more likely to hear a single sound:

- ~~Harmonic relationship~~
- Shared onset time
- Shared location
- Shared changes in amplitude (envelope)
- Shared changes in frequency (vibrato)

Compare to Gestalt principles of visual perception

Consonance & Dissonance

Rule of Thumb #4

Dissonance is caused by simultaneous frequencies that are close together

Two pitched sounds (sinusoids or complex waveforms) played simultaneously can be perceived as consonant or dissonant.

(not absolute binary, also has cultural dimensions)

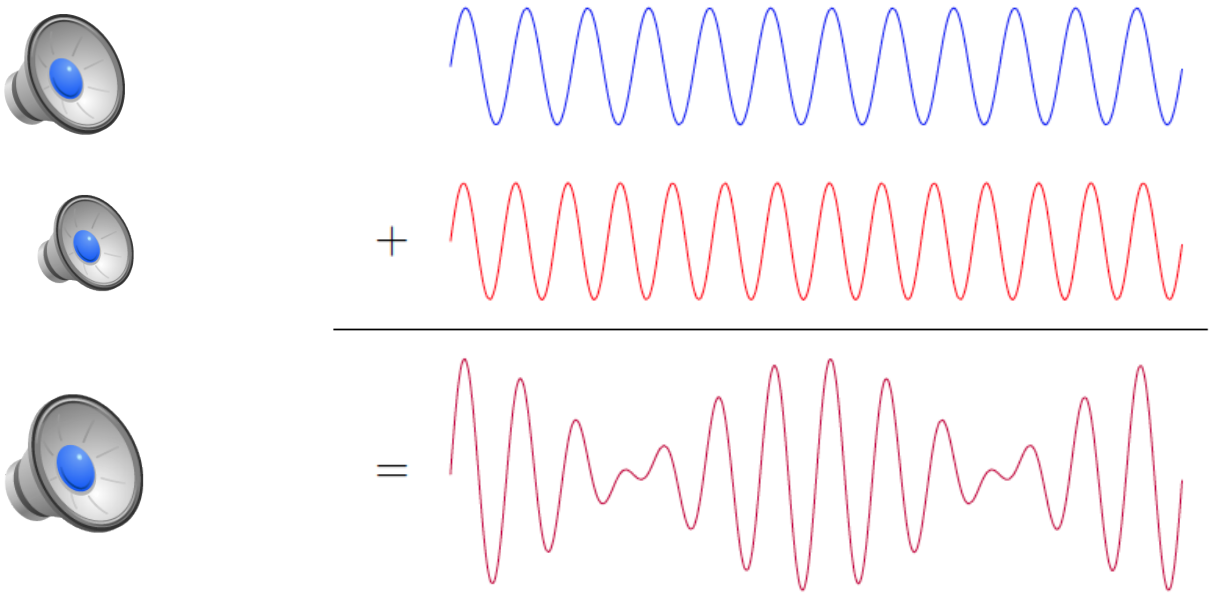
Perception impacted by:

- Relative pitch of sounds
- Absolute pitch of sounds
- Timbre of sounds

When frequencies are relatively close:

- Beating
- “Roughness”

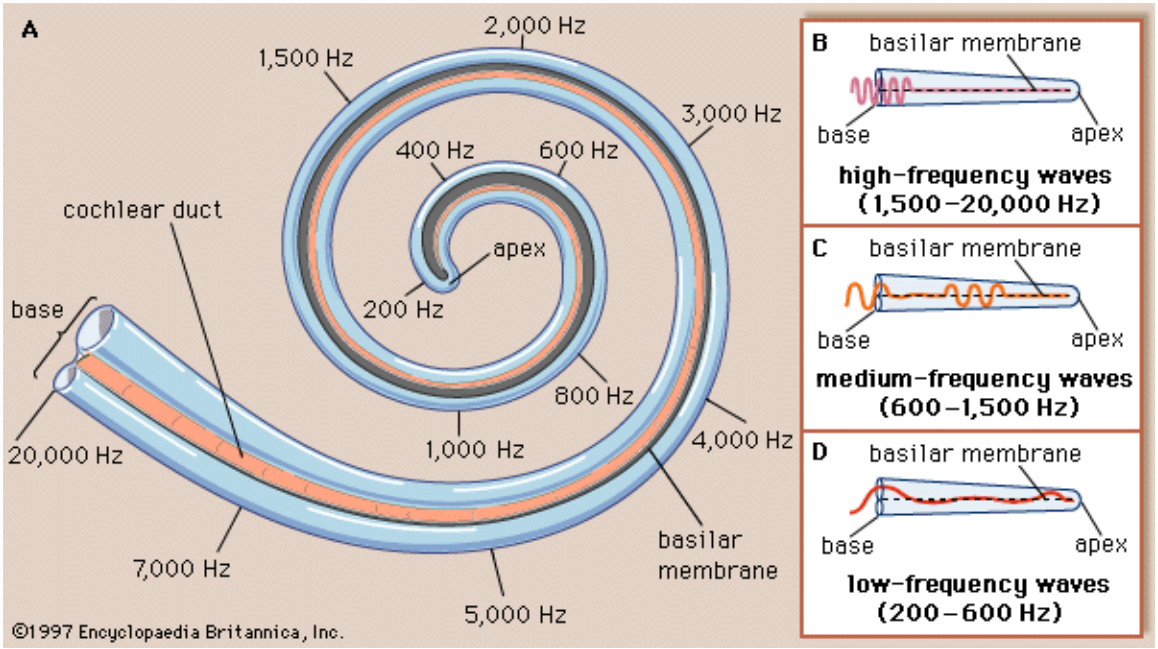
Beating (2 waves close in frequency)



$$\sin(A) + \sin(B) = 2\sin\left[\frac{A+B}{2}\right]\cos\left[\frac{A-B}{2}\right]$$

Audio examples

Basilar membrane



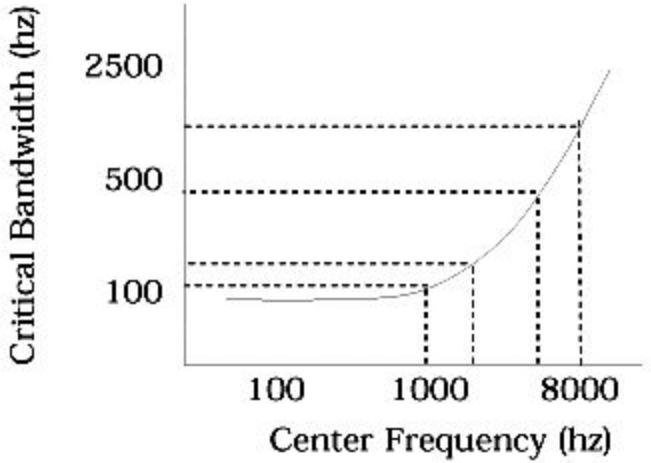
When two tones are close in frequency, they excite nearby locations on basilar membrane.

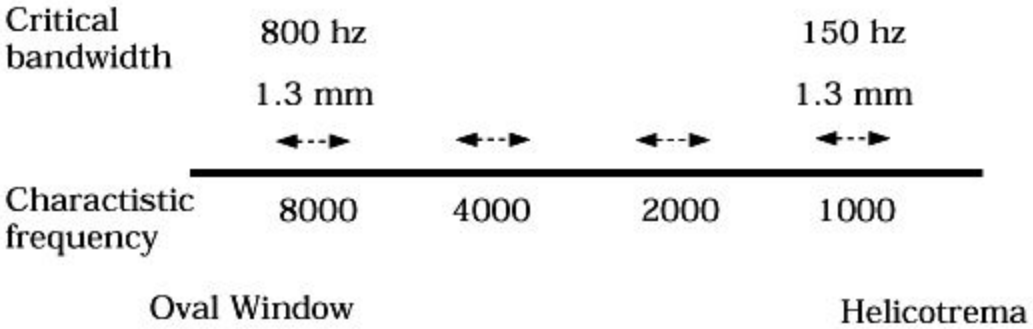
Critical band

A range of frequencies around a given tone within which addition of a second tone will interfere with accurate perception of the original tone.

Two simultaneous tones with different frequencies but within same critical band will sound “dissonant” or “rough.”

Critical bands and the basilar membrane





Demo

Two pitched sounds (sinusoids or complex waveforms) played simultaneously can be perceived as consonant or dissonant.

(not absolute binary, also has cultural dimensions)

Perception impacted by:

- Relative pitch of sounds
- Absolute pitch of sounds
- Timbre of sounds

When sounds aren’t just sinusoids

- Do harmonics/partials line up?
- Or do they fall within same critical bands, without lining up exactly?

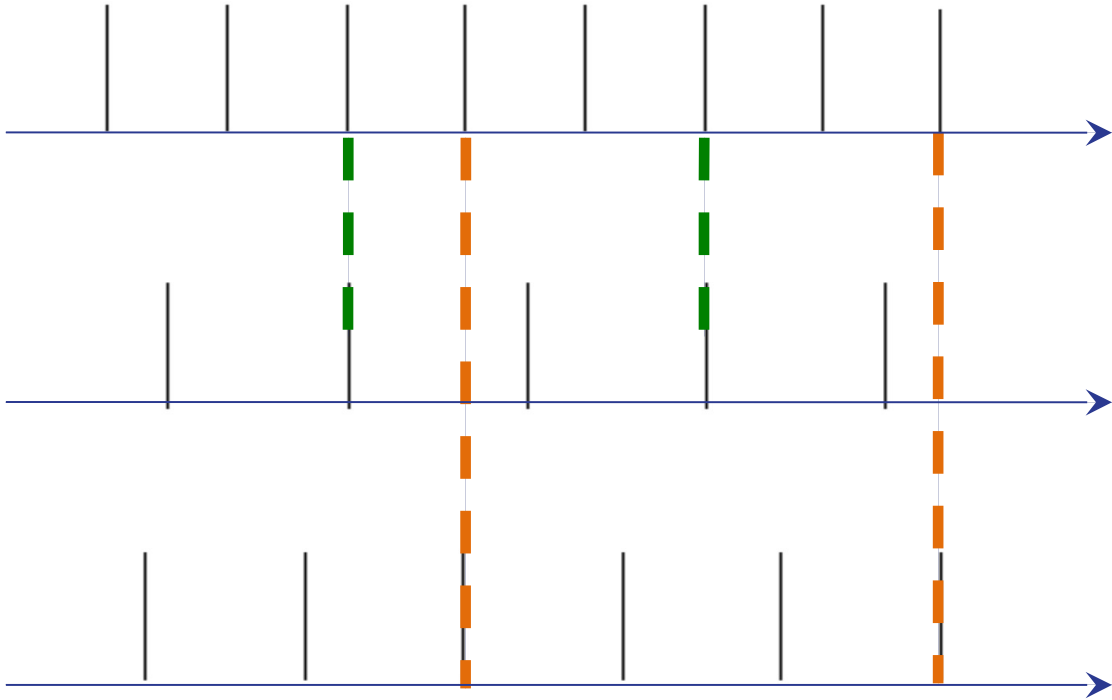
Demo

Consonant intervals reinforce each other

Fundamental f0 + harmonics

Fundamental (3/2*f0) + its harmonics

Fundamental (4/3*f0) + its harmonics



IMPORTANT FOR EXA

This is a text u take
the exam, do l about
signal percep soidal
decompositio better you
will do on the st exam
copies on the exams.
For example, otion
which are sim

Basic principle

We perceive it as unpleasant when our ability to accurately sense something is interfered with!

Speech analysis & perception

Human speech

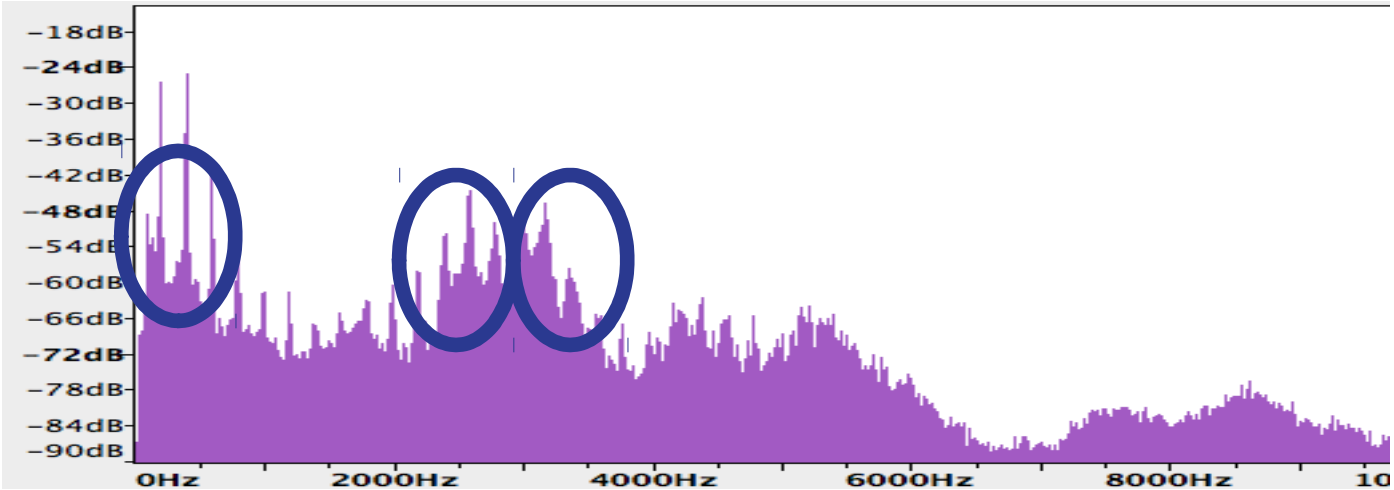
Listen to vowels: What do you hear?

- Constant pitch, volume
- Changing “tone quality”

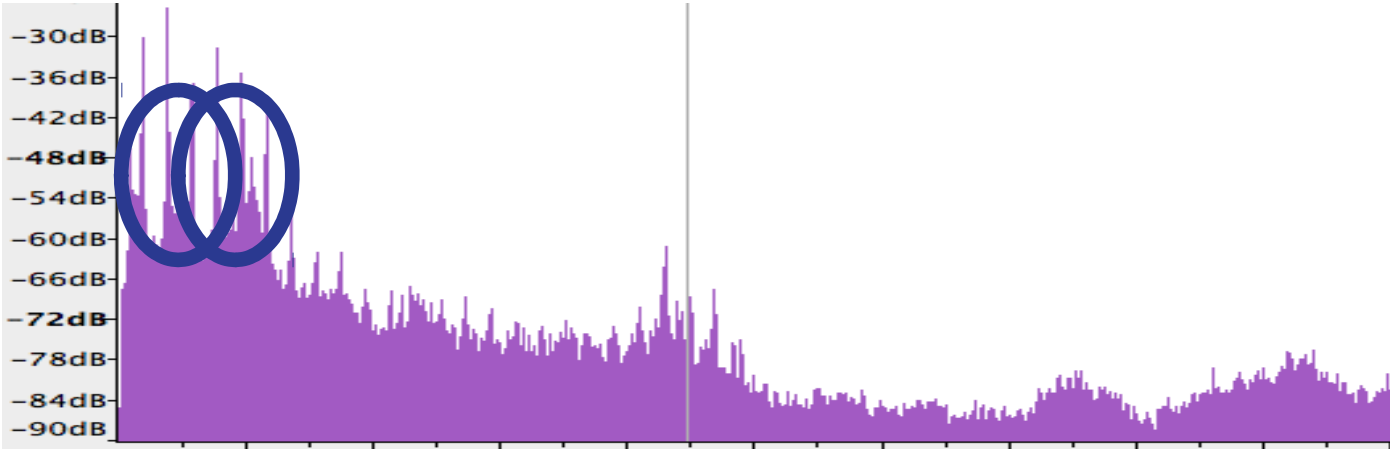
Rule of Thumb #5

Different vowels are distinguished by relative strengths of particular frequency ranges (“formants”)

EEEE


-18dB
-24dB
-30dB
-36dB
-42dB
-48dB
-54dB
-60dB
-66dB
-72dB
-78dB
-84dB
-90dB
0Hz
2000Hz
4000Hz
6000Hz
8000Hz

OOOO


-30dB
-36dB
-42dB
-48dB
-54dB
-60dB
-66dB
-72dB
-78dB
-84dB
-90dB

Formants

Different vowels exhibit greater magnitude in different regions of the frequency spectrum.

Formants

The first two formants are sufficient to distinguish vowel sound.

Consonants

No definite pitch
(unperiodic, inharmonically related partials)
Still distinguishable by frequency content

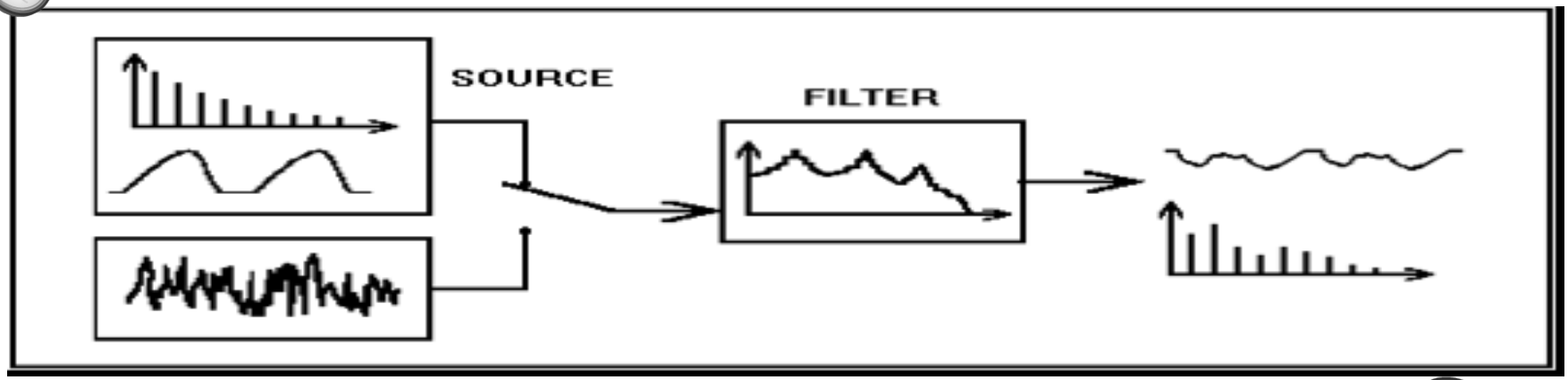
Demo: sndpeek

Singer’s Formant

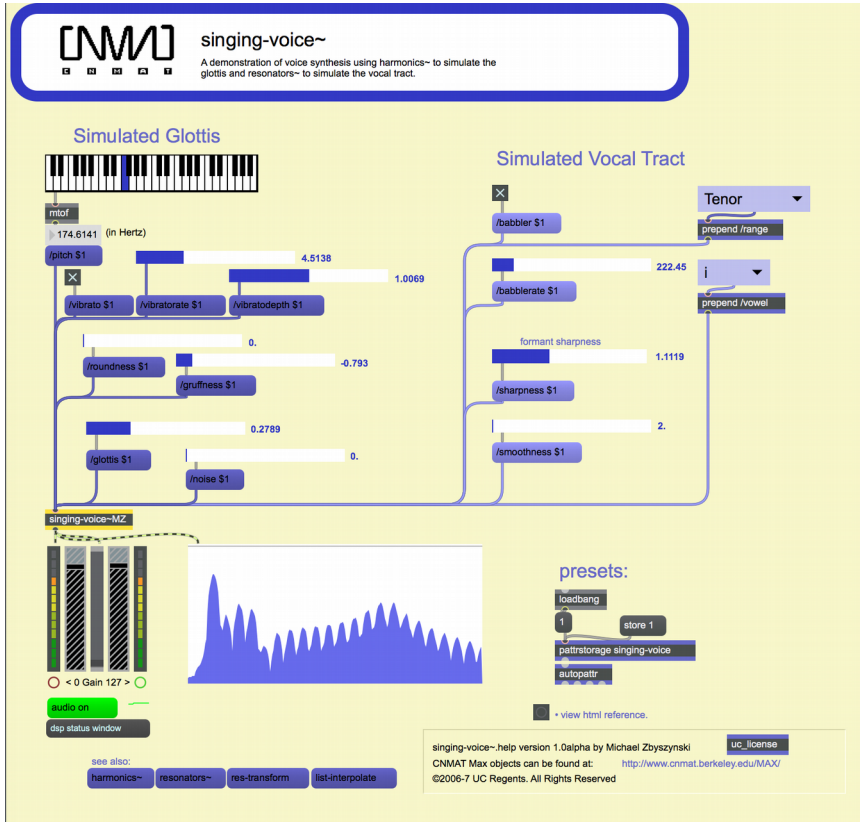
Trained singers have additional formant around 3000Hz.

That allows a singer to be heard above orchestra!

Source-filter model of voice synthesis & analysis



Singing voice demo

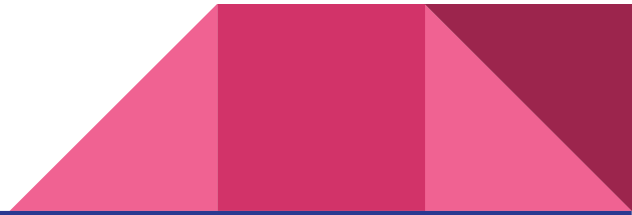


Cross-synthesis

Demo:
guitar source spectrum shaped by voice spectrum

Practical applications

- Speech synthesis
- Speech as spectral manipulation
- Compression
- Auto-tune



Speech perception also has a visual component

Demo: McGurk effect

http://www.youtube.com/watch?v=jtsfidRq2tw

Fourier Analysis

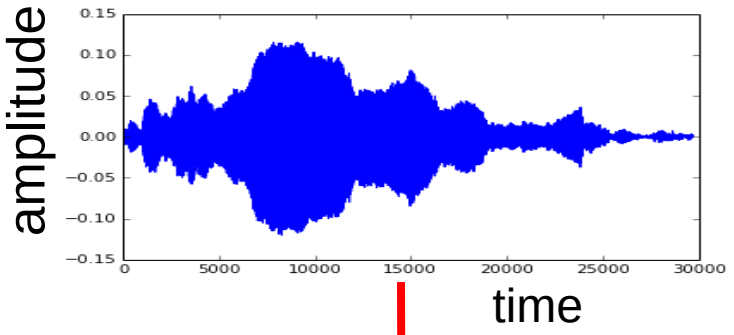
=


CON ANIMA
White Wine
VINEGAR

sound

1D array of amplitudes

1 amplitude value for each sample point in time

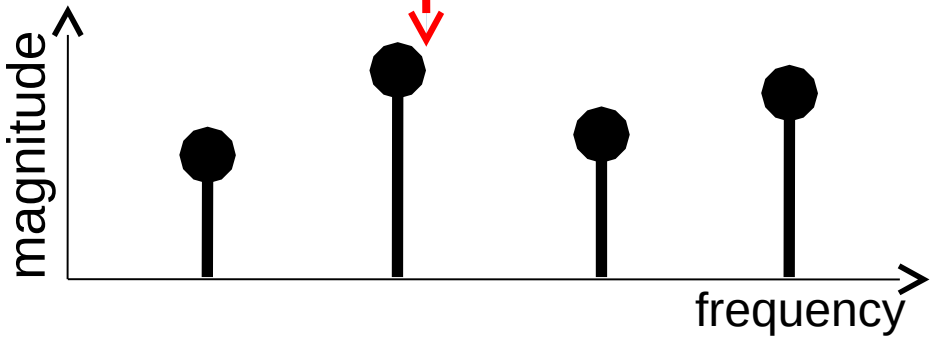


Fourier Transform

spectrum

1D array of spectrum bins

- how much of each frequency?
- and at what phase? (not pictured)



image

2D matrix of brightness

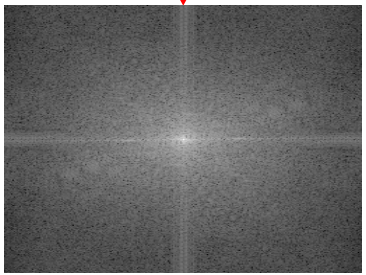
1 brightness for each (x,y) pair

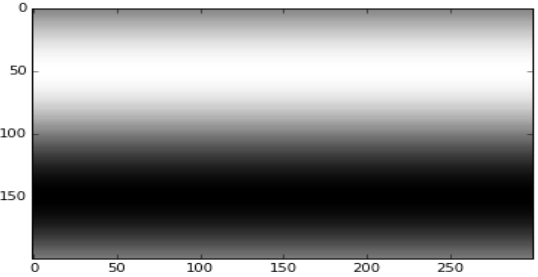
Fourier Transform

spectrum

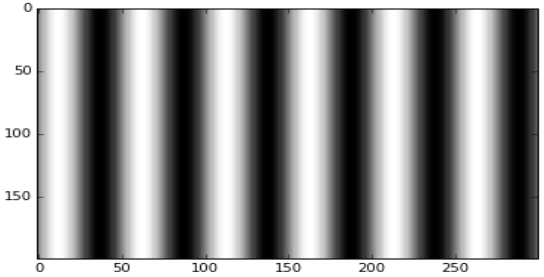
2D matrix of spectrum bins

- how much of sine at each frequency and **angle**?
- and at what phase? (not pictured)

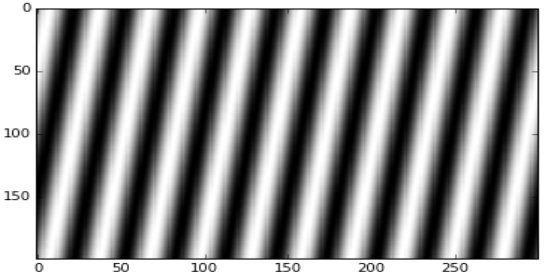



0
50
100
150
0
50
100
150
200
250

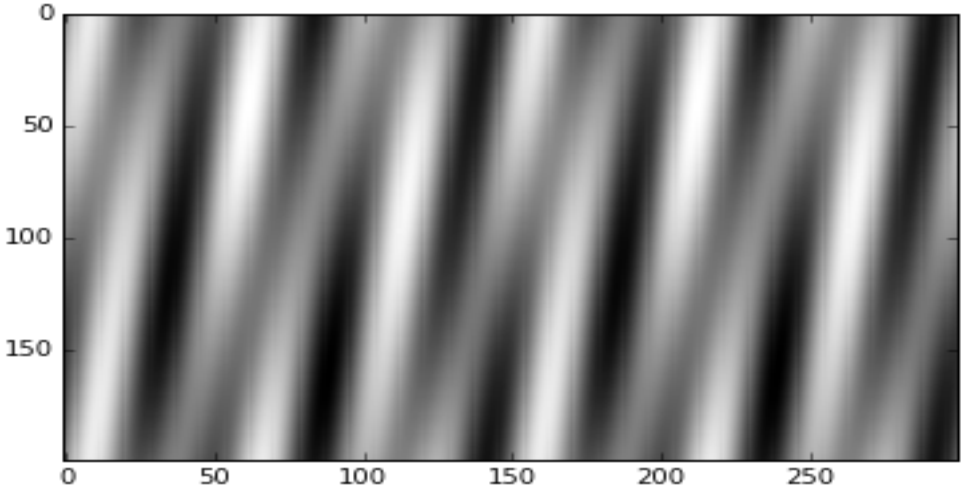
+


0
50
100
150
0
50
100
150
200
250

+


0
50
100
150
0
50
100
150
200
250

=


0
50
100
150
0
50
100
150
200
250

sound

1D array of amplitudes

N samples long

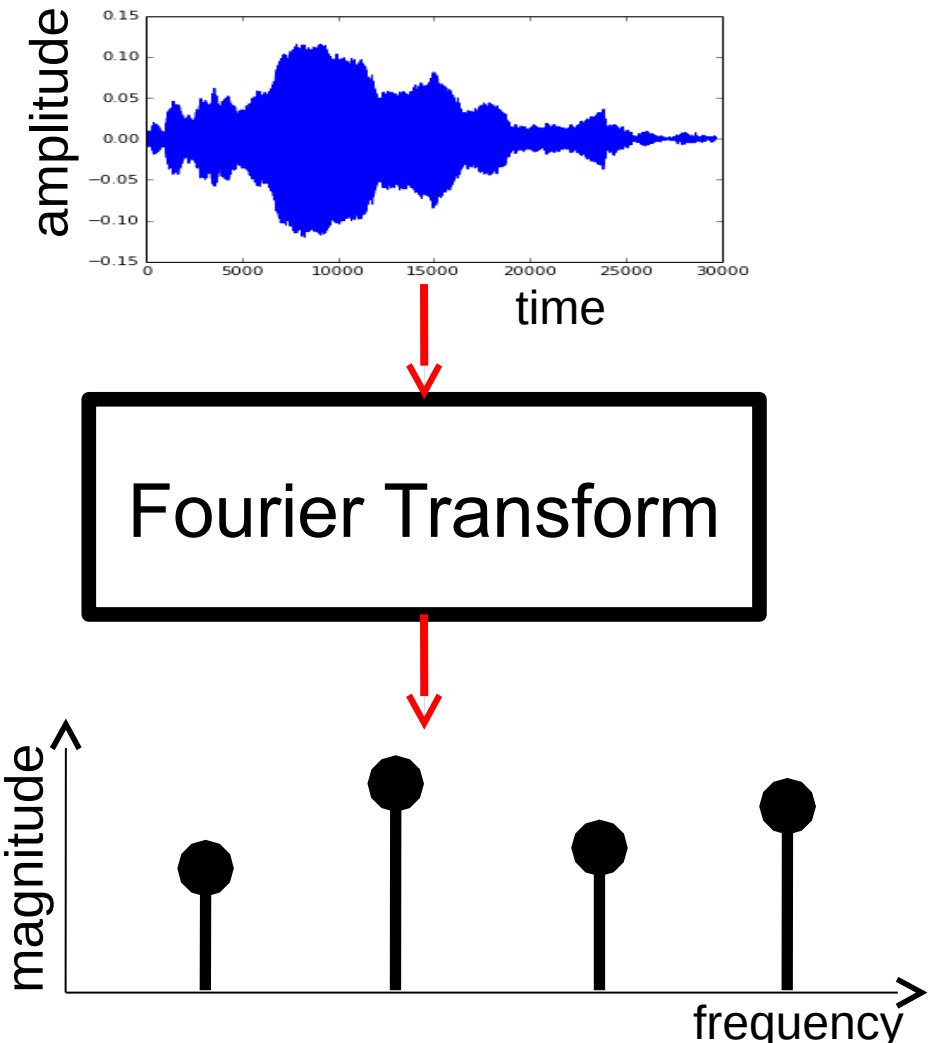
from time t=0 to N-1

spectrum

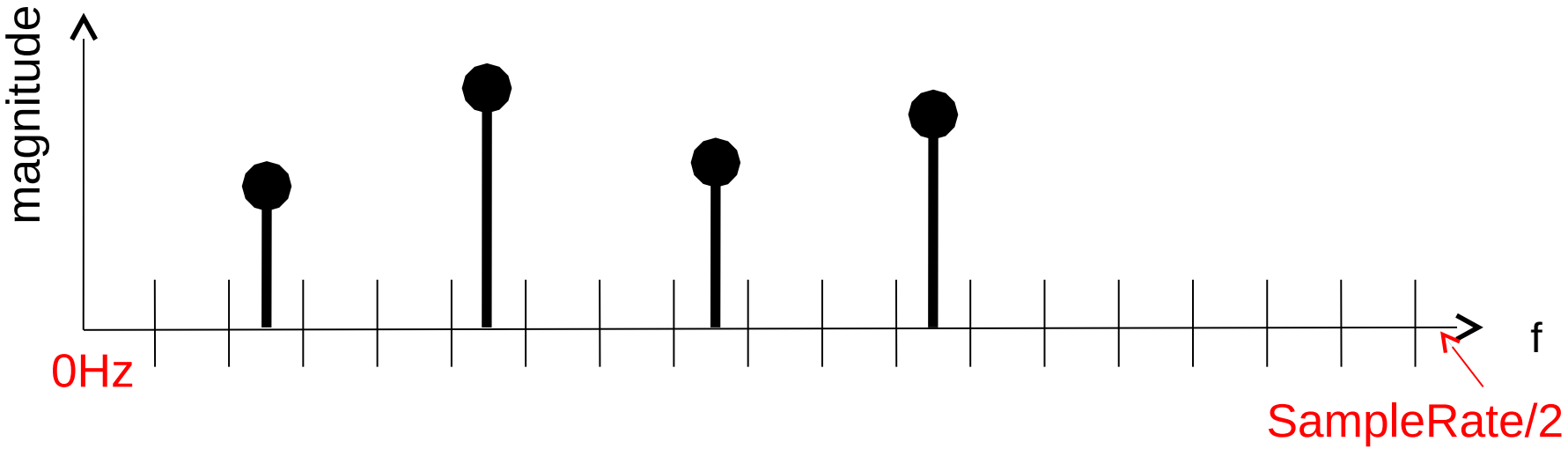
1D array of spectrum bins

N bins long

from 0Hz to Sample Rate



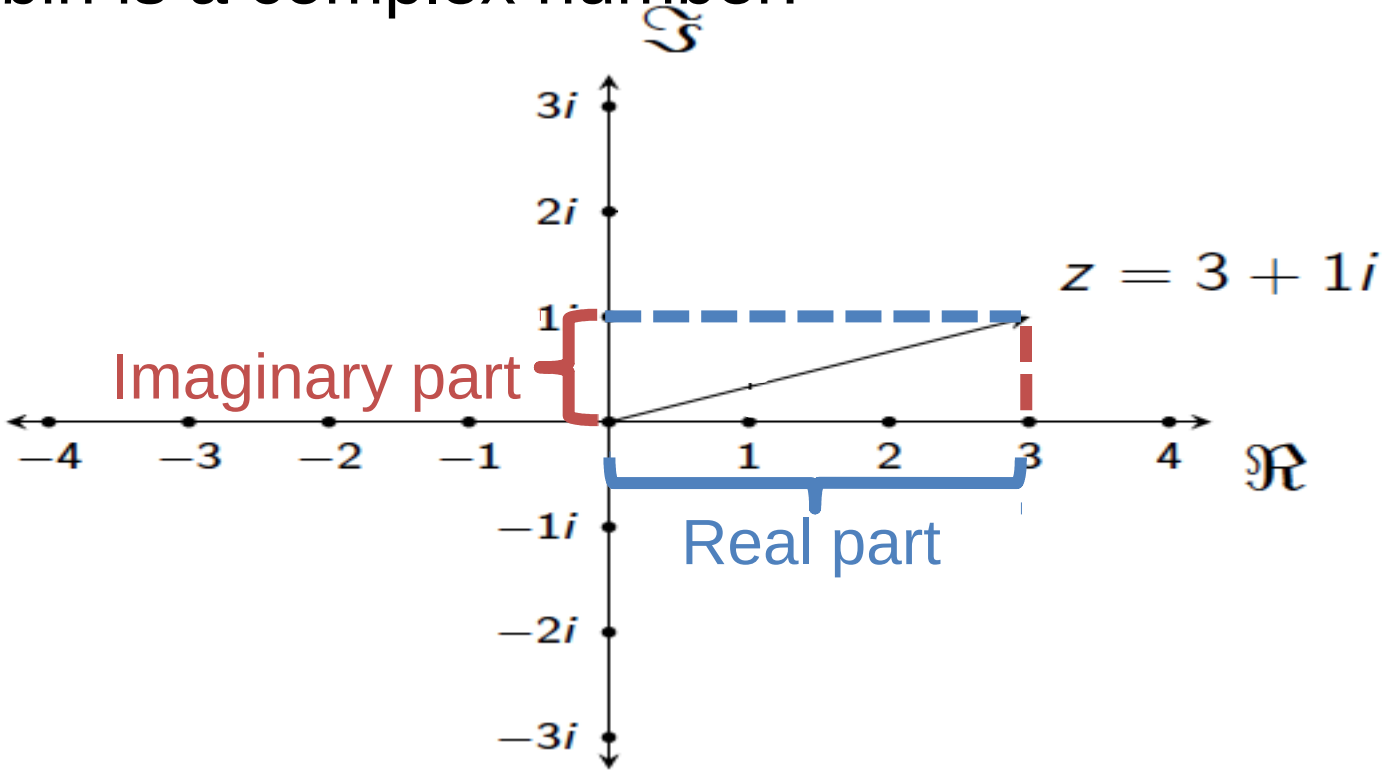
An N-sample signal can be reconstructed by adding N/2 sinusoids together!



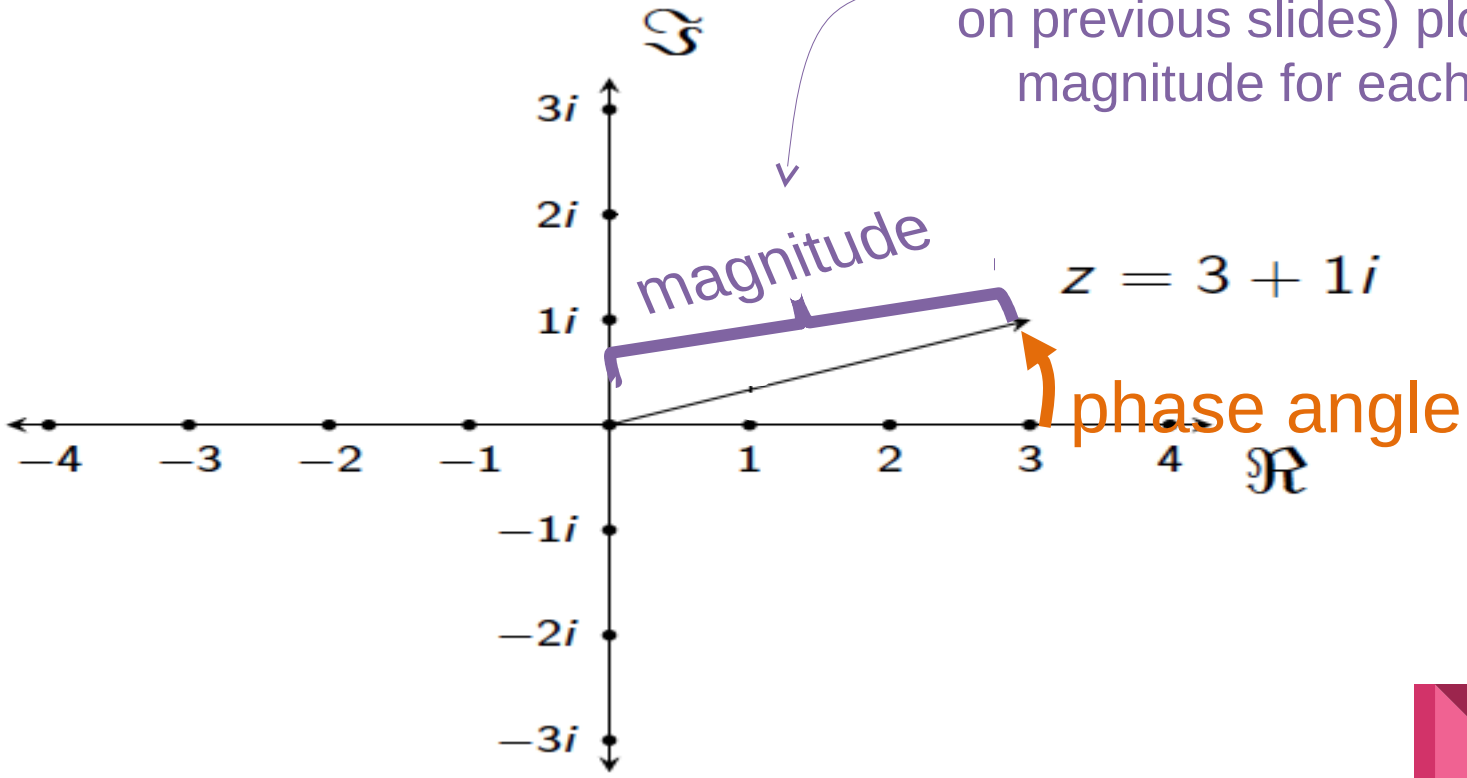
Evenly chop frequencies from 0Hz to Nyquist into N/2 bins

→ Bin width is (SampleRate / 2) / (N/2) = 1/N * SampleRate

Each bin is a complex number:



A magnitude spectrum (like on previous slides) plots the magnitude for each bin



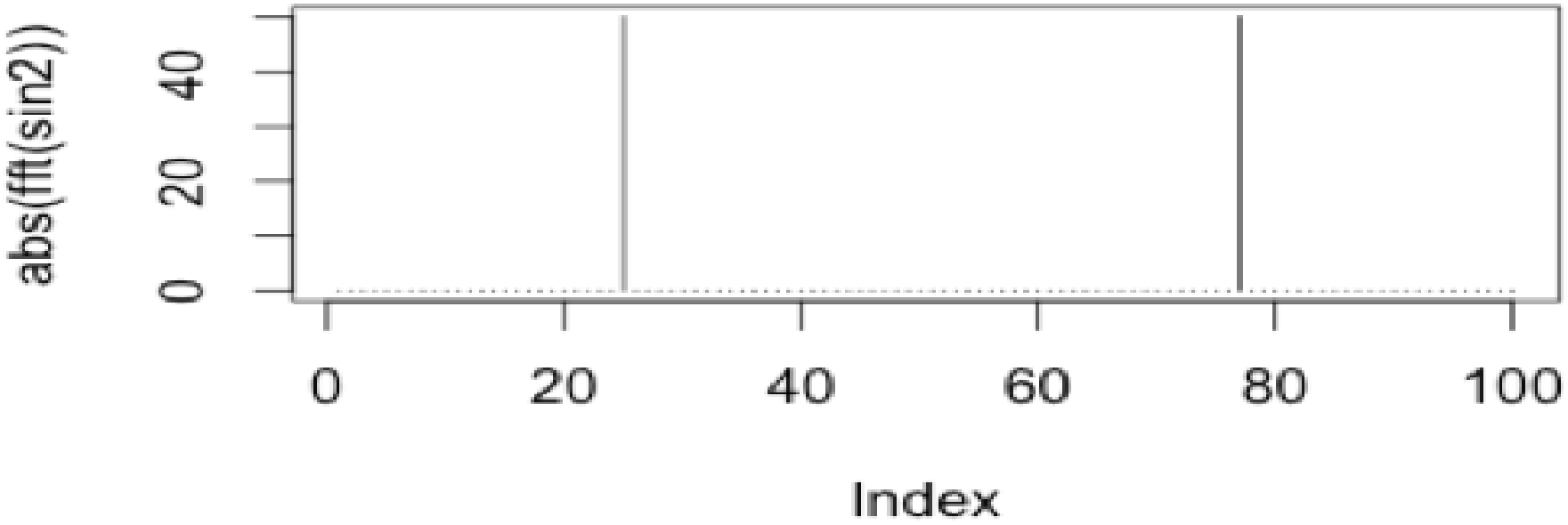
Summary: to discover what frequencies are present in a signal

1. Compute the Fourier Transform of the signal
   - If you compute spectrum on N samples (e.g., 100 samples), you will get back a spectrum that has N *bins*.
   - This will tell you about *what* frequencies are present in these N samples, but not *when*!
2. Keep bins 0 to N/2 and throw away the rest.
   - This discards everything above the Nyquist frequency (higher bins will duplicate the lower half of the spectrum anyway)
3. Each bin is a complex number; for plotting and many other analysis tasks, we just care about the magnitude (in Python and some other languages, can compute using abs())

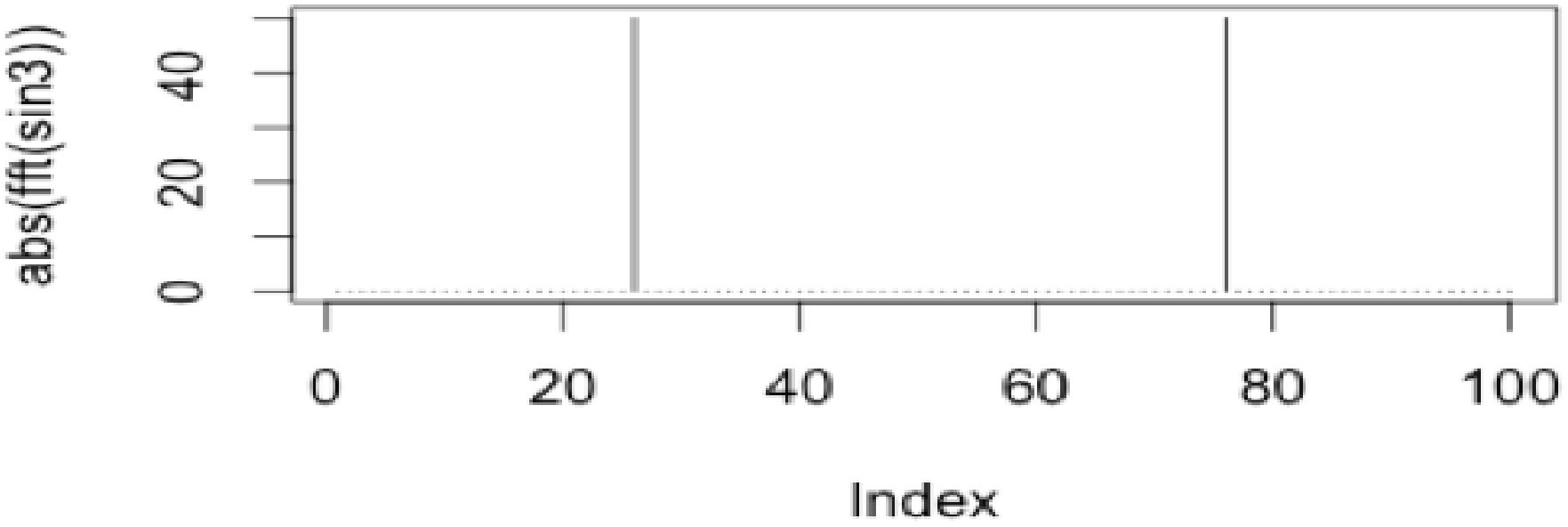
What happens if my signal contains a frequency that’s not exactly equal to the center frequency of a bin?

This frequency will “leak” into nearby bins.

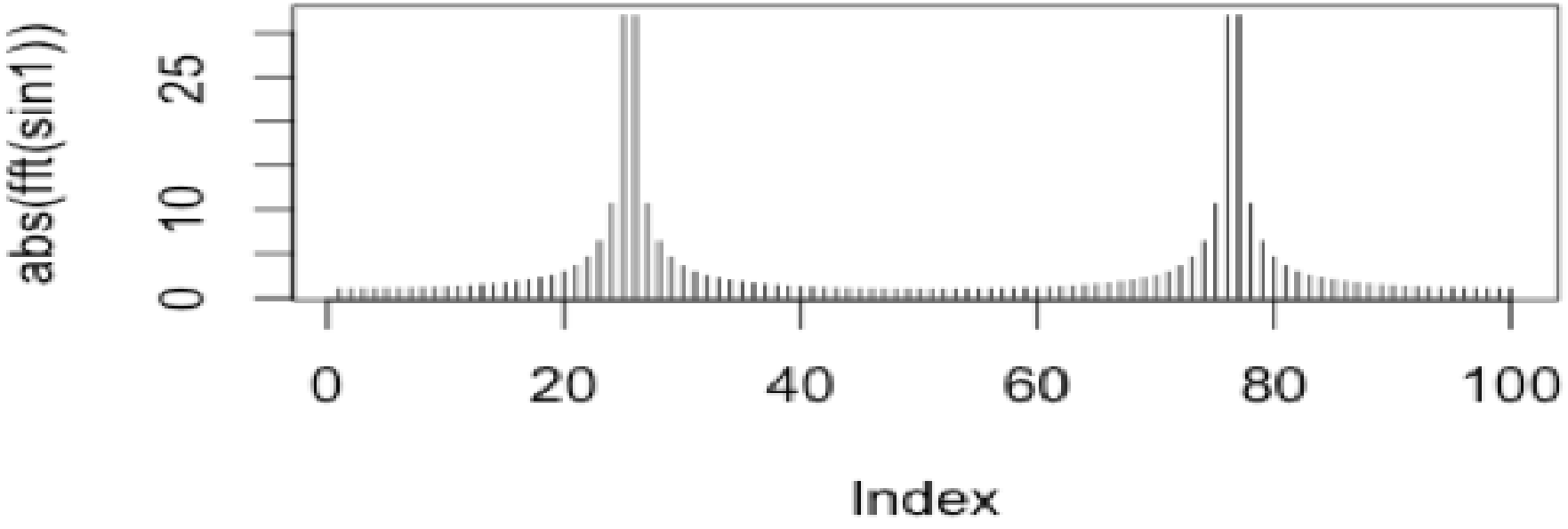
SR = 100Hz, sine at 24 Hz



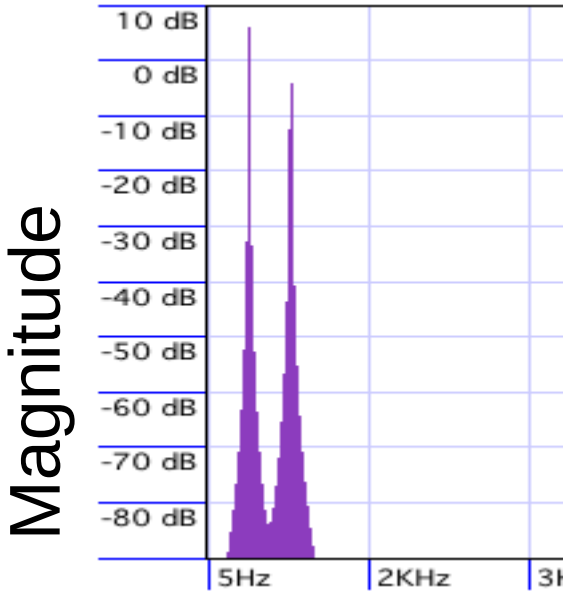
SR = 100Hz, sine at 25 Hz

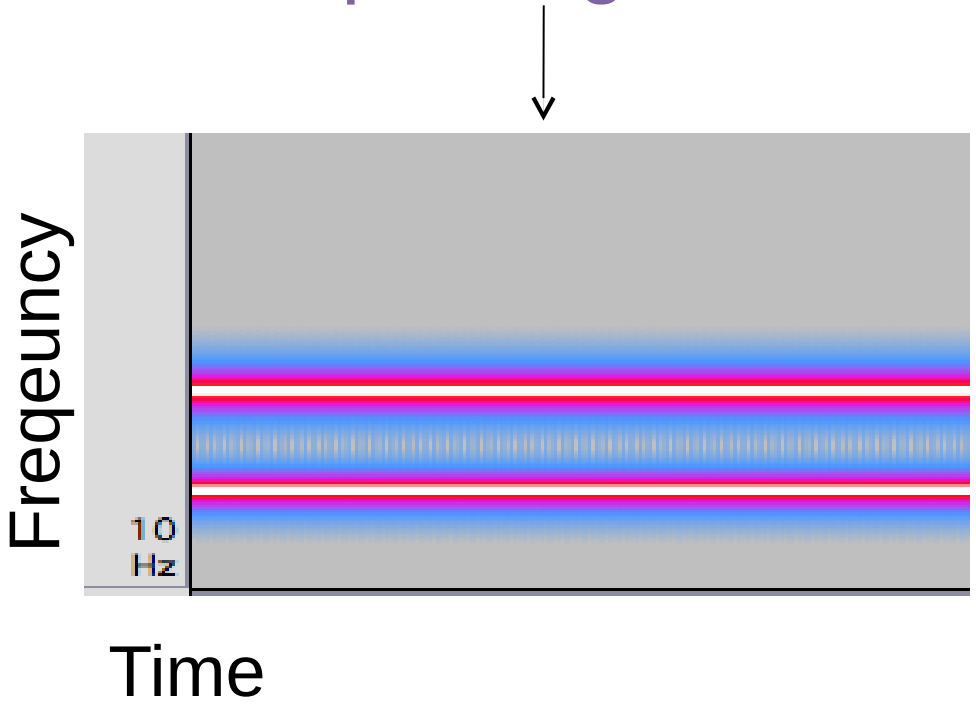


SR = 100Hz, sine at 24.5 Hz



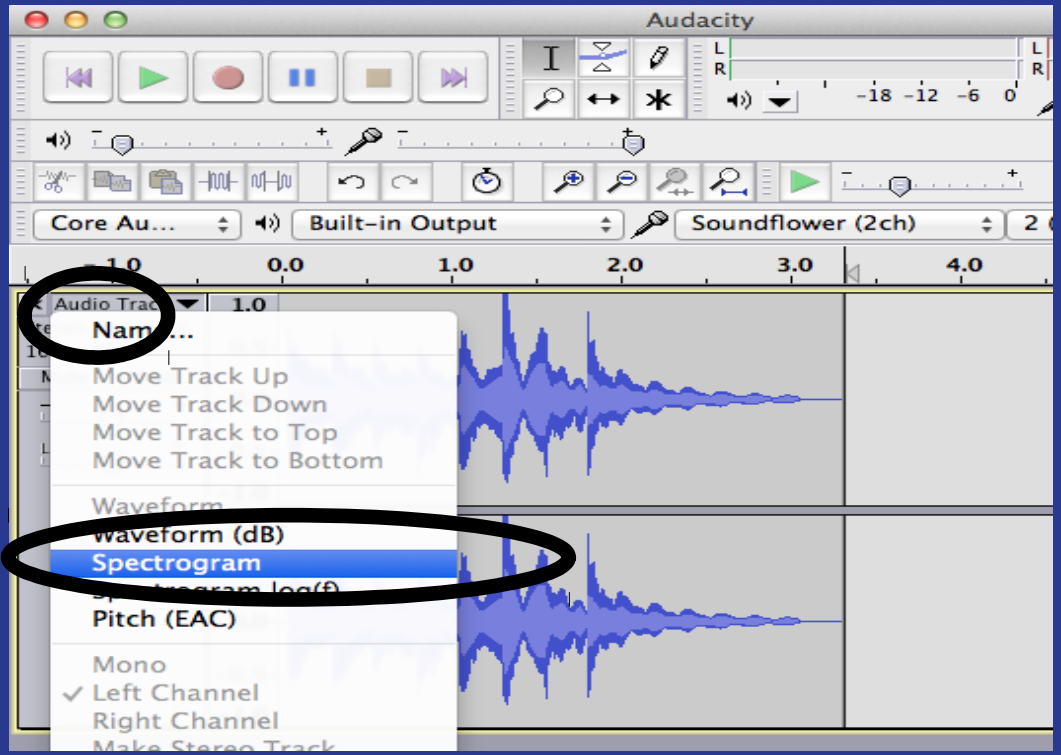
Visualise FFT over time as spectrograms





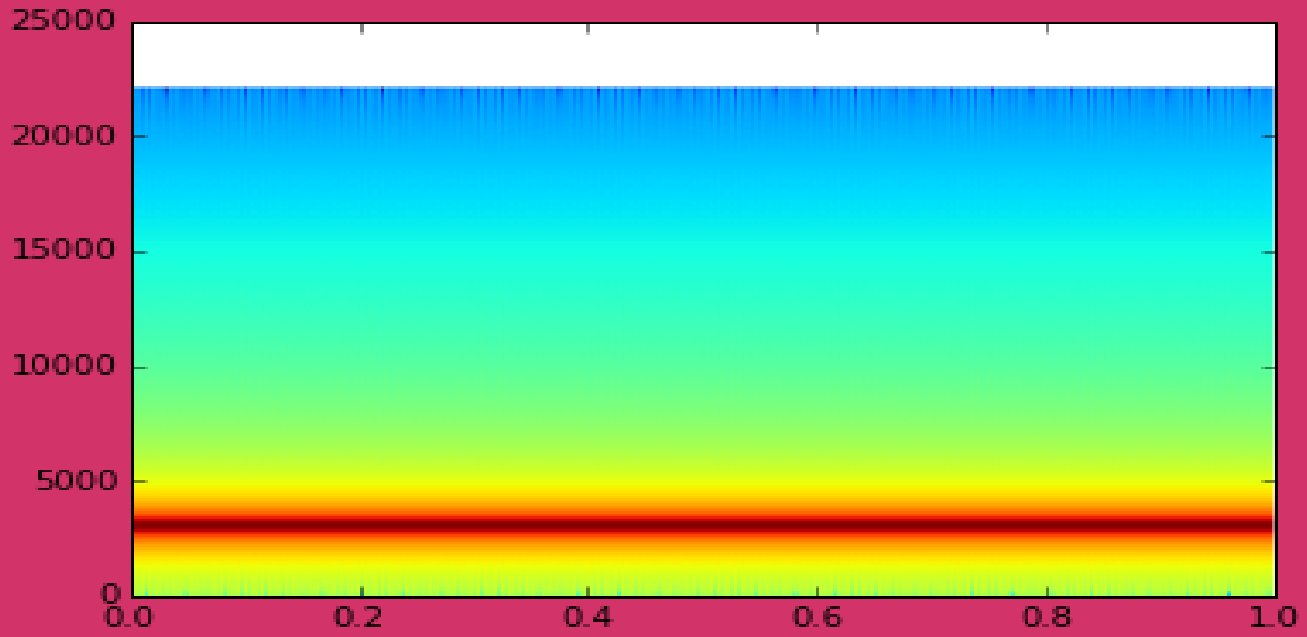
Color = magnitude

In Audacity

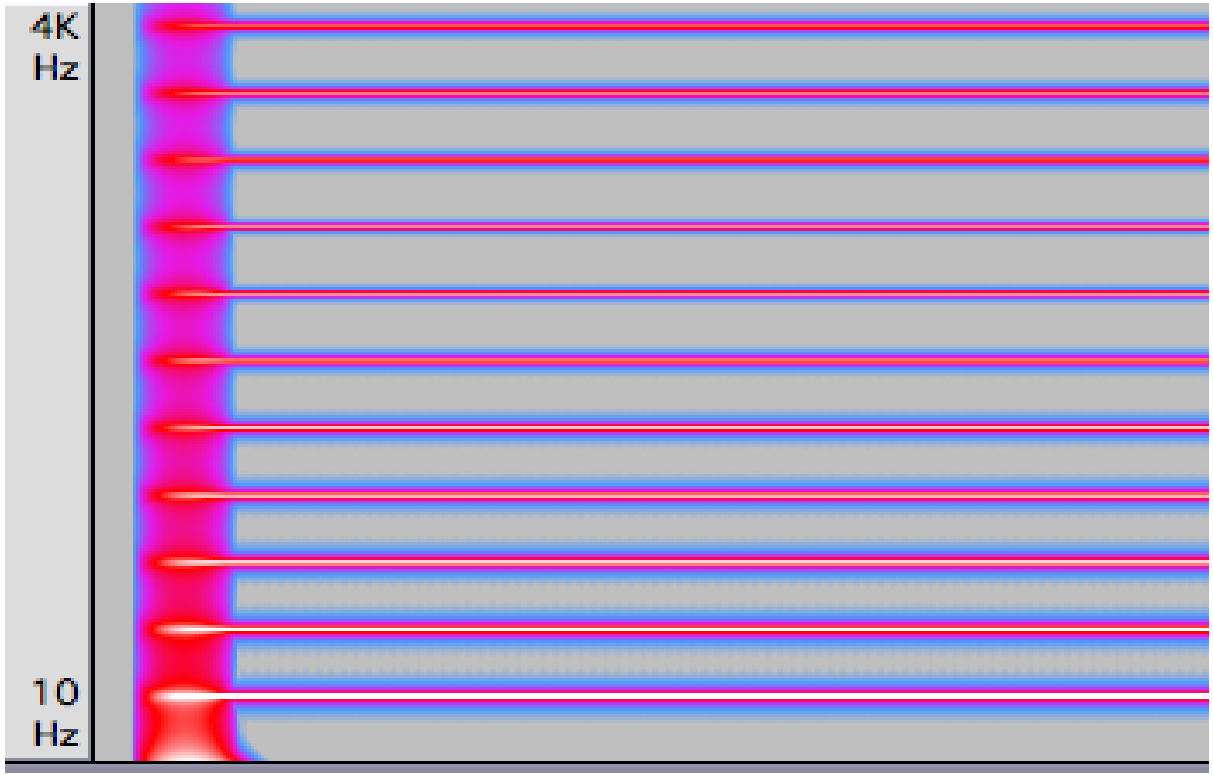

Audacity
Core Au...
Built-in Output
Soundflower (2ch)
Move Track Up
Move Track Down
Move Track to Top
Move Track to Bottom
Waveform
Waveform (dB)
Spectrogram
Pitch (EAC)
Mono
Left Channel
Right Channel

In iPython

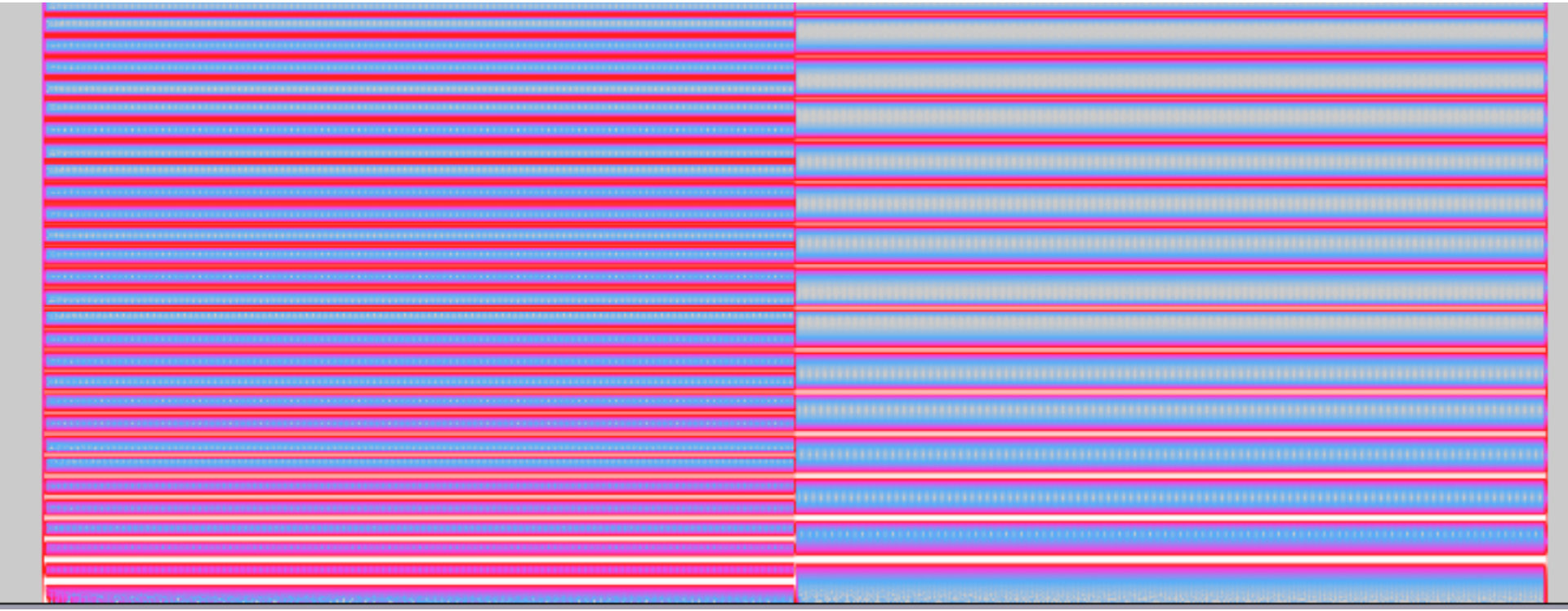
```
t = linspace(0, 1, 44100)
s = sin(2*pi*3000*t)
sp = specgram(s, Fs=44100)
```



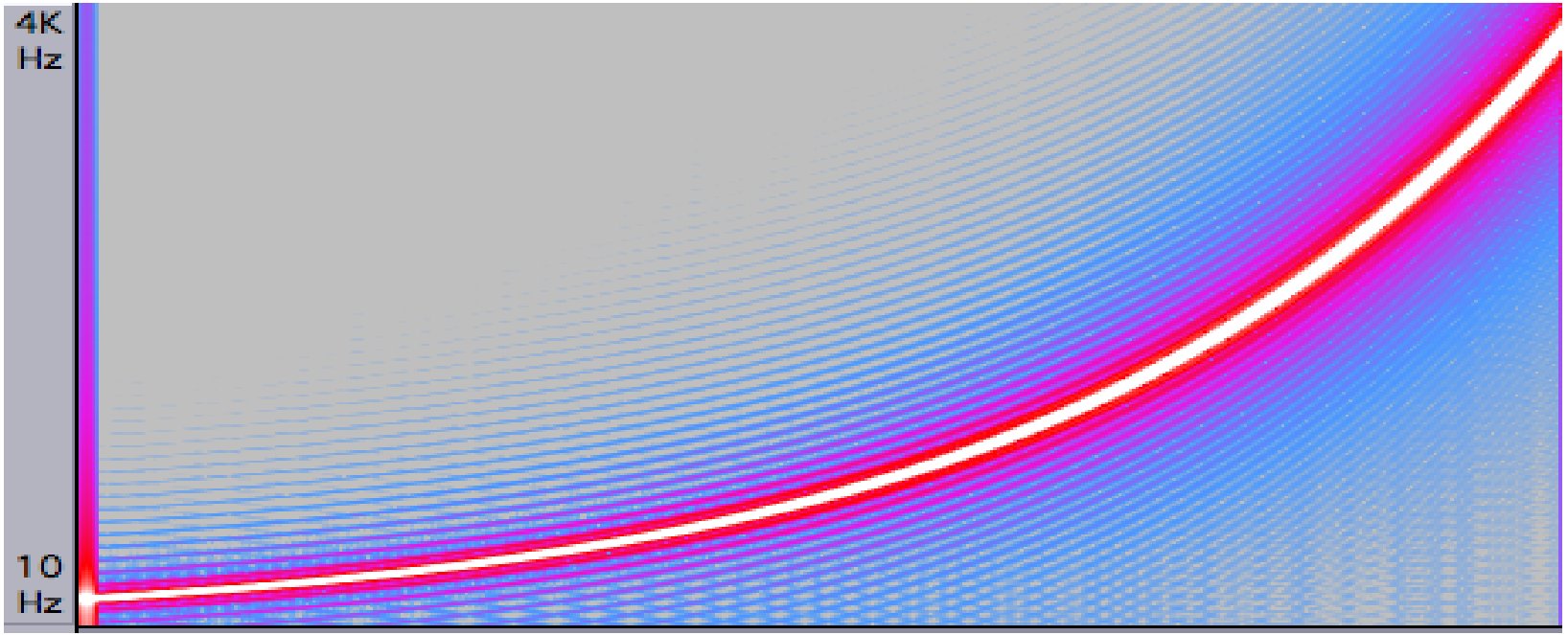
What will you hear?



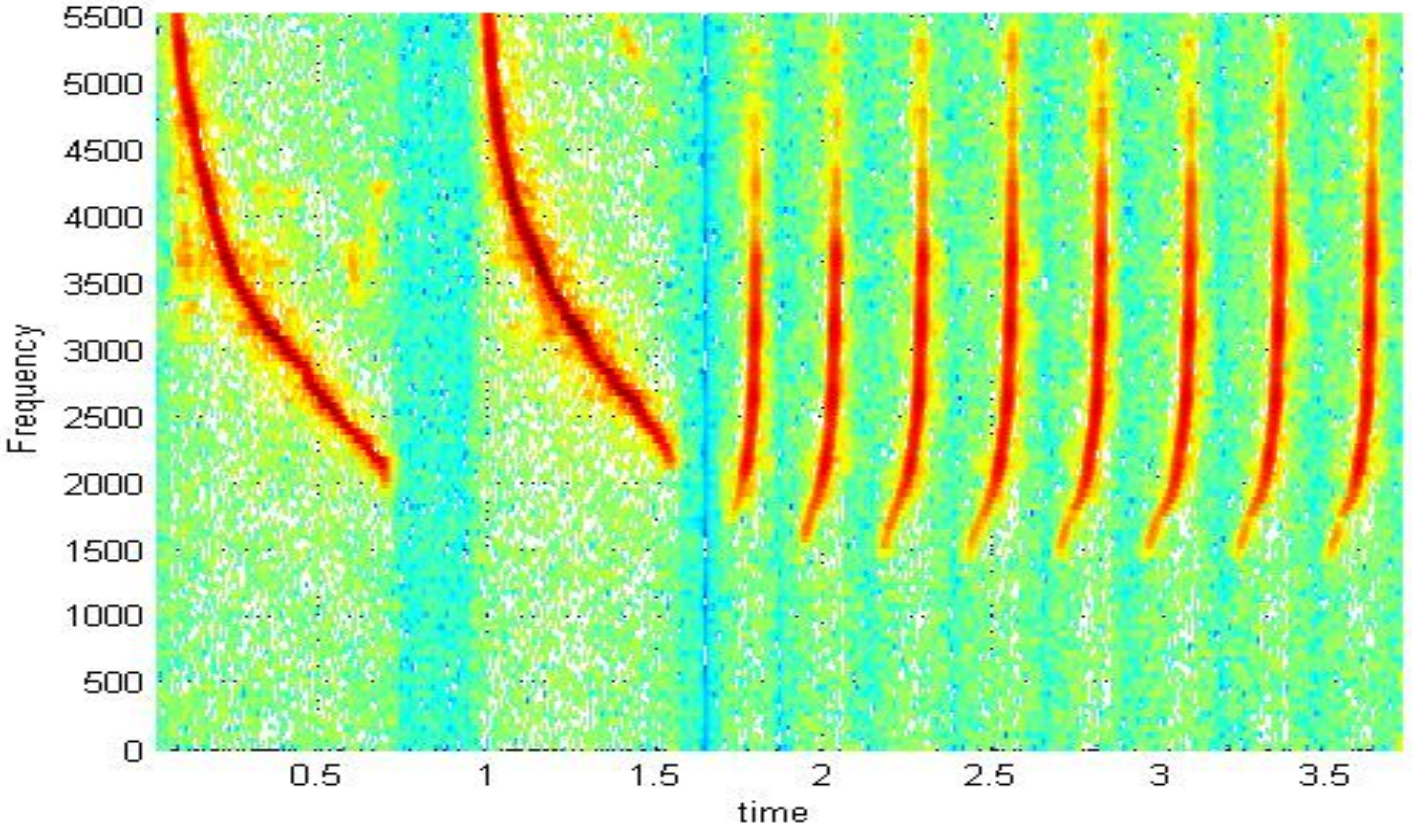
What will you hear?



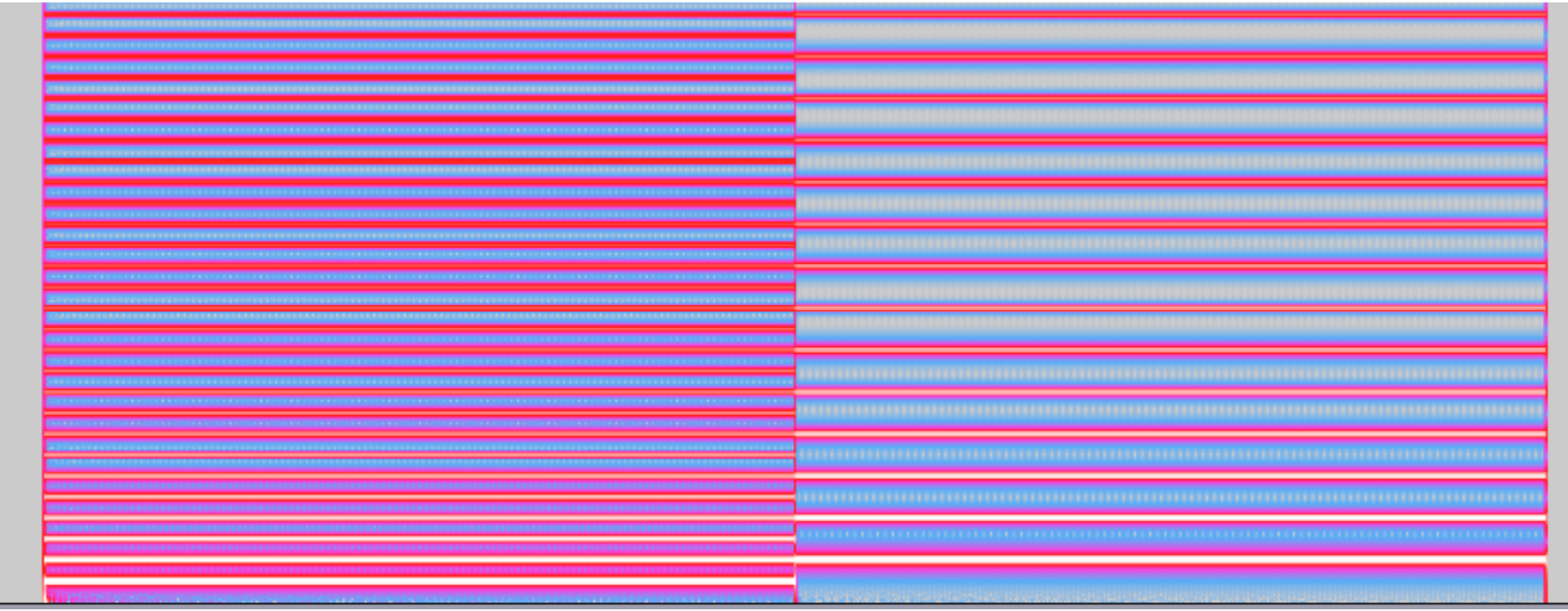
What will you hear?



What will you hear?



What will you hear?



How many bins to use? (What should *N* be?)

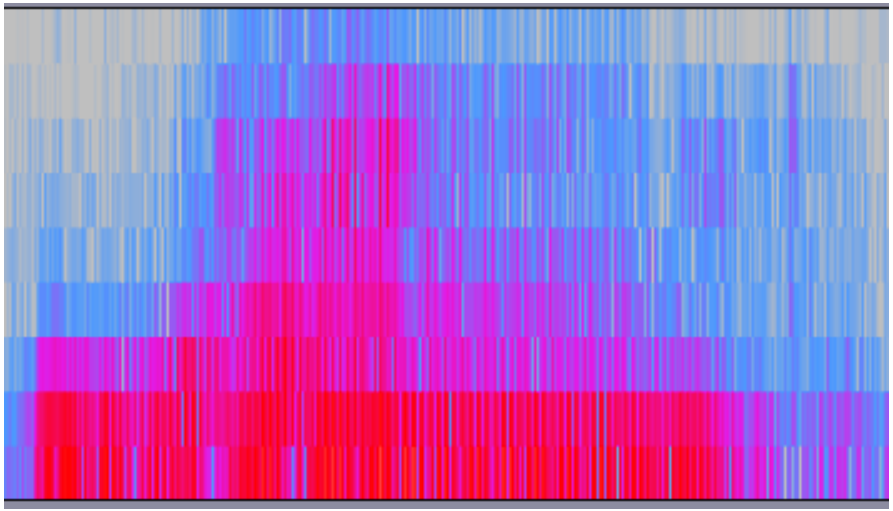
More bins?

- Better frequency resolution
- Worse time resolution (FFT can’t detect changes within the analysis frame)

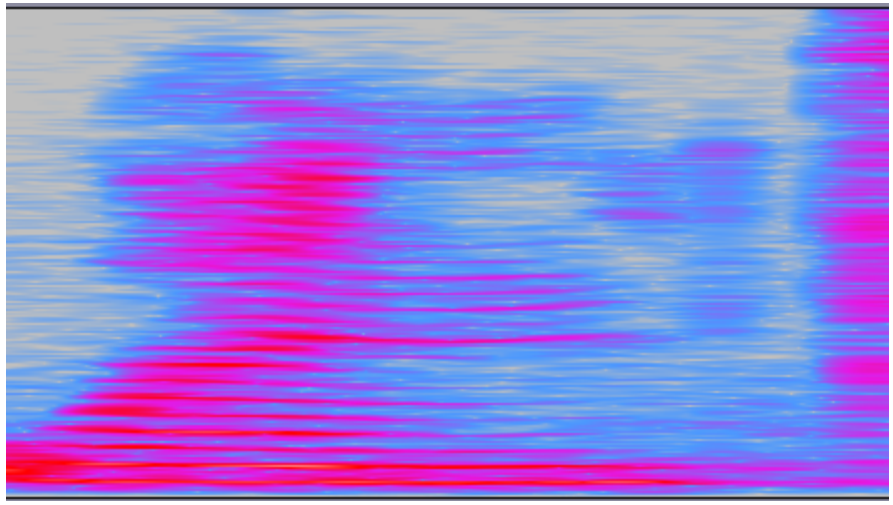
Fewer bins?

- Worse frequency resolution
- Better time resolution

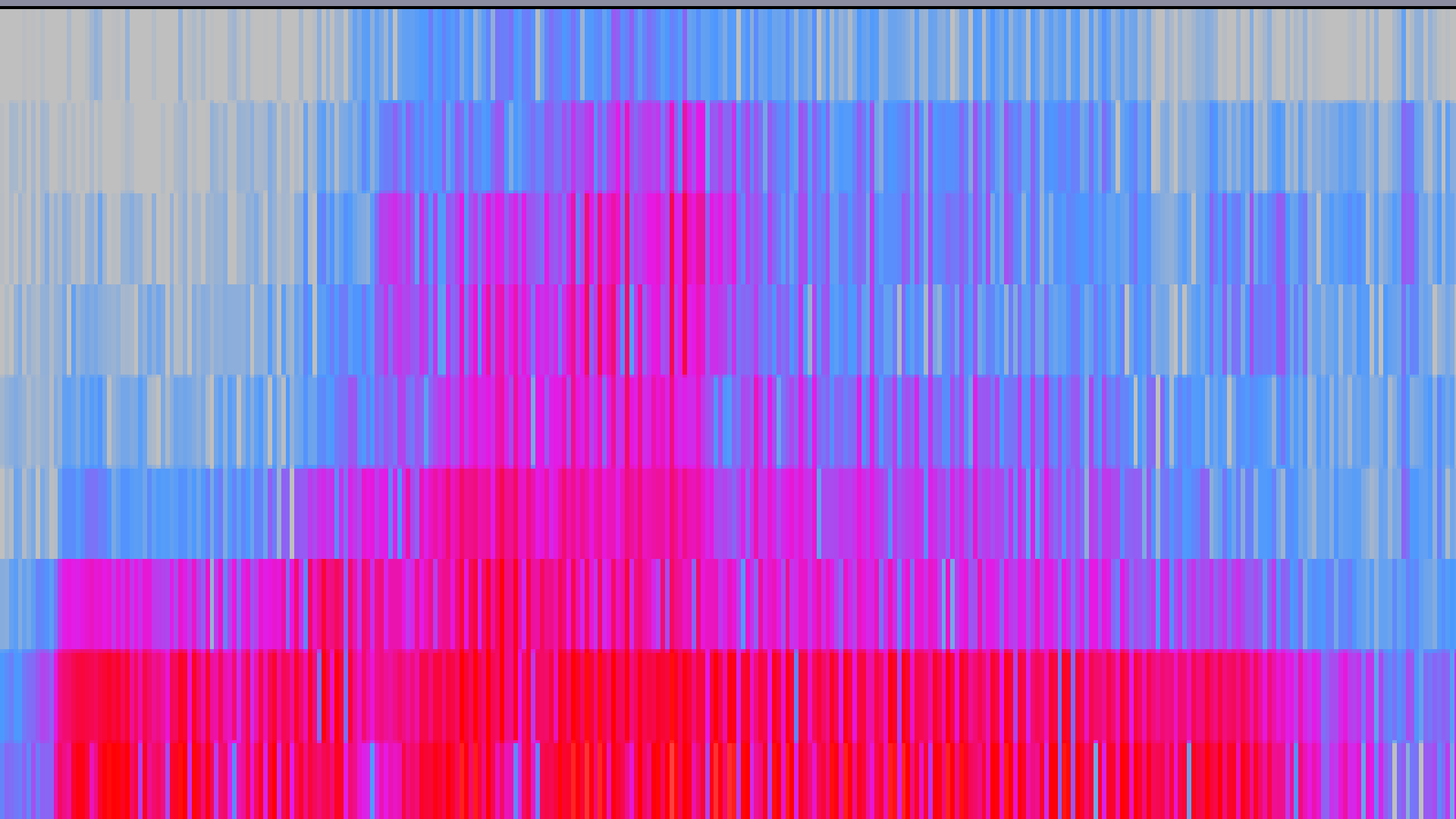
Time/Frequency tradeoff

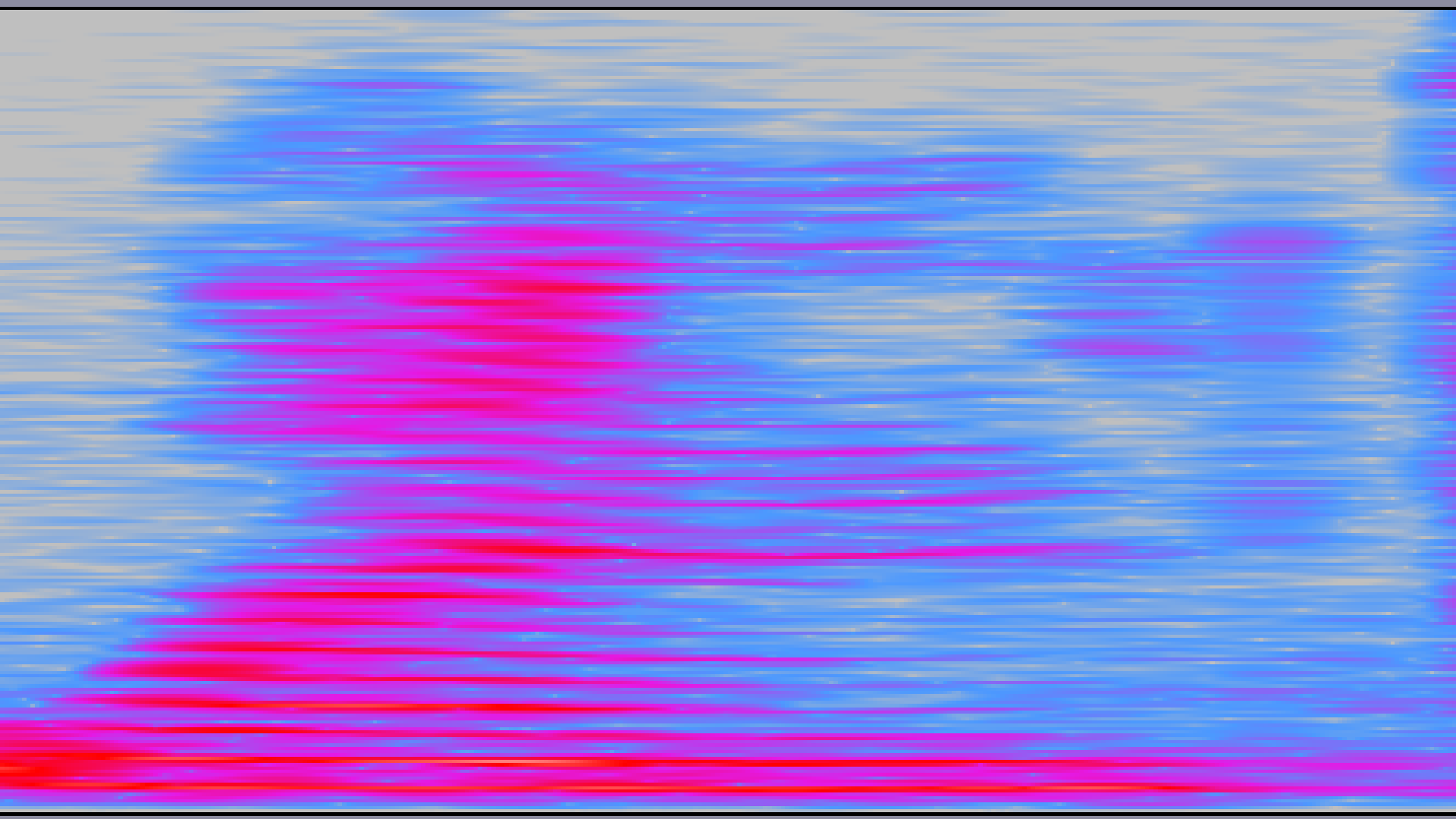


N=64



N=4096





How does the FFT discover what frequencies are present in a signal?

Key principles from last term:

1) All media signals are functions

2) All functions can be expressed as sums of sinusoids

Sinusoids are “basis functions” →
You can add together carefully-chosen sinusoids to get any function

Intro to Fourier Analysis

Given a waveform, what are its sinusoidal components?

If $s(t) = A_1 \sin(2\pi f_1 t + \Phi_1)$

$+ A_2 \sin(2\pi f_2 t + \Phi_2)$

$+ A_3 \sin(2\pi f_3 t + \Phi_3)$

$+ \ldots$

The spectrum of s

What are A_n, f_n, Φ_n for all n?

1. Determining amplitude A_n for basis function at frequency f_n

Let x[n] be the n^{th} sample of x

$s_k[n]$ be the n^{th} sample of the basis sine at frequency k

and N be the length (# samples) of x

Compute:

$x[1] * s_k[1] + x[2] * s_k[2] + \ldots + x[N] * s_k[N]$

Handling phase

Recognize that a sinusoid with non-zero phase is just a weighted sum of a sine and cosine at the same frequency!

$$\sin(2\pi ft + \Phi)$$

$$=$$

$$\cos(\Phi) * \sin(2\pi ft) + \sin(\Phi) * \cos(2\pi ft)$$

The Discrete Fourier Transform

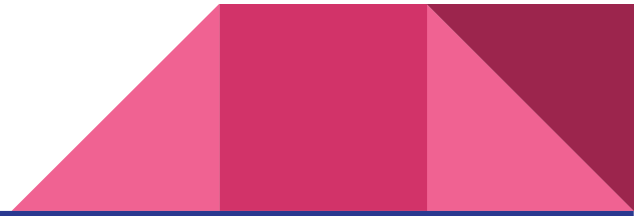
$$X_k = \sum_{n=1}^{N} x[n] \times e^{-i 2\pi (k/N) n}$$

x is our input signal (e.g. our audio signal)

Discussion: Computation time

Recall: Summation is like a for-loop in code

We do *N* summations, each with *N* iterations

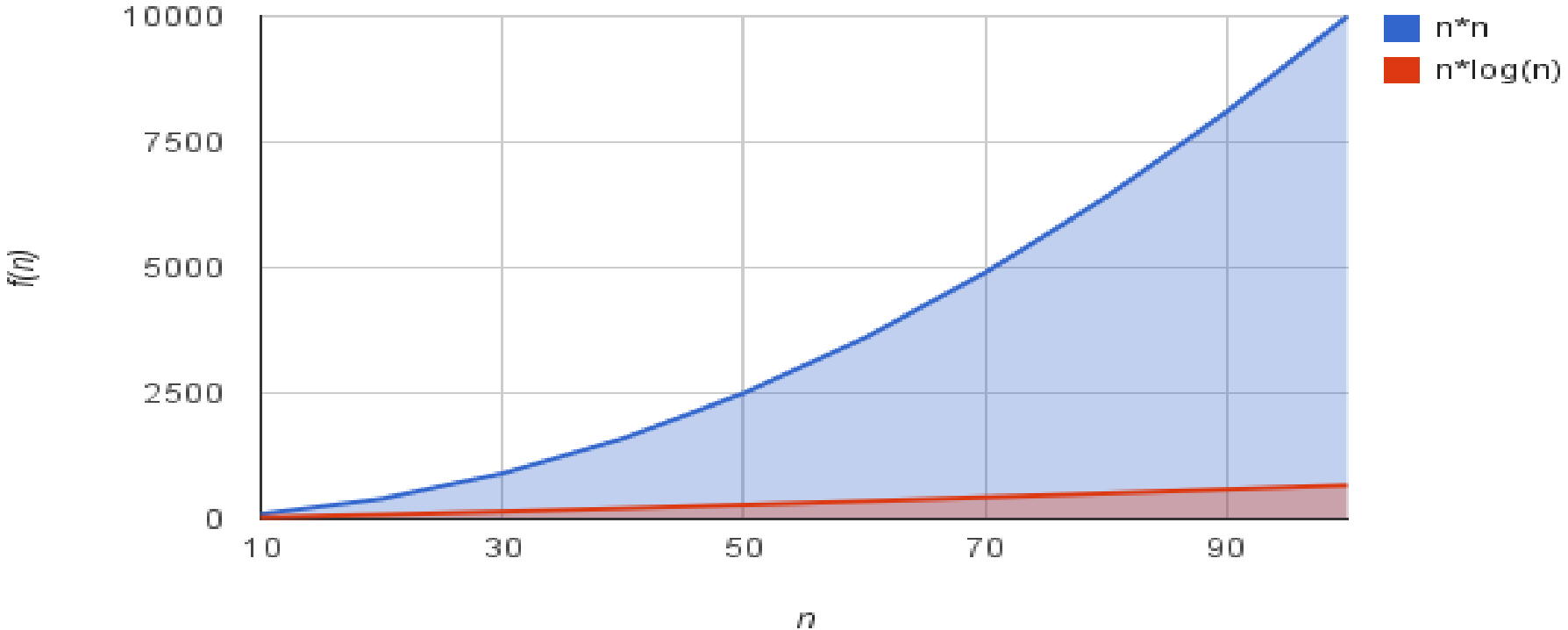


A faster way

The FFT :

- FAST Fourier transform ☺
- Computes the same results, but using fewer computations
- O(N log N) rather than $O(N^2)$
- Fastest when N is a power of 2

$O(N^2)$ vs $O(N \log N)$



From http://java.dzone.com/articles/algorithm-week-insertion-sort

Why Fourier Analysis?

Why Fourier analysis?

Audio:

- Tells us about pitch, timbre, instrumentation, mastering; speech/speaker; recording environment; …
- Re-synthesize and process sounds (e.g. time stretch, pitch shift)
- Reason about how filters, reverb, EQ, etc. will affect a sound
- *Design* filters, reverb, EQ, etc.