Eng. 100: Music Signal Processing
DSP Lecture 5
Lab 3: Signal Spectra (Continued)

Curiosity (major to minor conversion)

- https://www.youtube.com/watch?v=oQPZikexR1Y (macabre macarena)
- https://www.youtube.com/watch?v=LbTxfN8d2CI (don't worry, really?)
- https://www.youtube.com/watch?v=rGflu3TbREo (whymca)
- https://www.youtube.com/watch?v=aouKyslYMMg (roar)
- https://www.youtube.com/watch?v=uxfNmLW7v0o (viva la vida)

Announcements:

- Study Lab 3 carefully *before* lab!
- HW2 due Friday

Outline

- What: The spectrum of a signal (first class)
  - Part 1. Why we need spectra
  - Part 2. Periodic signals $x(t) = c_0 + \sum_{k=1}^{K} c_k \cos\left(2\pi \frac{k}{T} t - \theta_k\right)$
  - Part 3. Band-limited signals, $K = \lfloor BT \rfloor$
- How: Methods for computing spectra (second class)
  - Part 4. Sampling rate $S > 2B$
  - Part 5. By hand by solving systems of equations
  - Part 6. Using general Fourier series solution
  - Part 7. Using fast Fourier transform (FFT), *e.g.*, in `Julia`
- Why: Using a signal's spectrum (third class)
  - to determine note frequencies: $f = \frac{k}{N} S$
  - to remove unwanted noise
  - to visualize frequency content (spectrogram)
  - Lab 3

Learning objectives

- Understand $K = \lfloor BT \rfloor$ for band-limited, periodic signals
- Understand sampling requirement: $S > 2B$
- Understand spectra of band-limited, periodic signals
- Compute Fourier series coefficients $\{a_k, b_k\}$ from signal samples
- Compute spectra using FFT
- Understand FFT output (determine frequency via FFT)
- Relate line spectra to stem plots of FFT

Summary of previous class

- $\arccos$ method for finding frequencies is not robust enough (non-sinusoids, noise, polyphony)
  $\Longrightarrow$ need better frequency estimation methods to transcribe
- Signal properties:
  - T-periodic signals: $x(t) = x(t+T)$ for "all" t
  - band-limited signals: maximum frequency B
- Fourier series representations of T-periodic signals:

$$x(t) = c_0 + c_1 \cos\left(2\pi \frac{1}{T} t - \theta_1\right) + c_2 \cos\left(2\pi \frac{2}{T} t - \theta_2\right) + \cdots$$

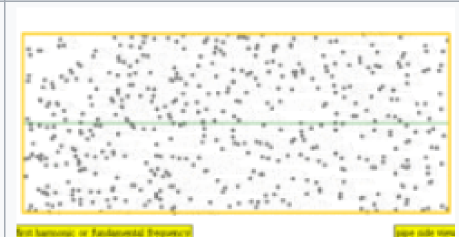
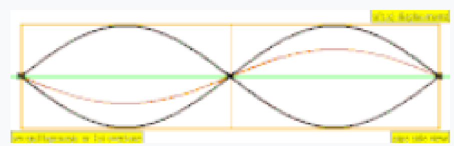
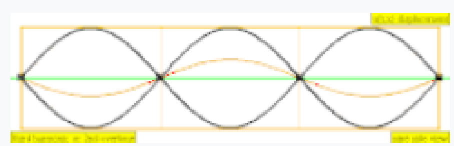
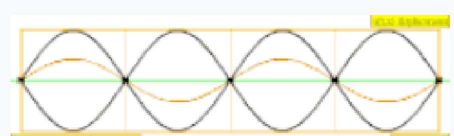
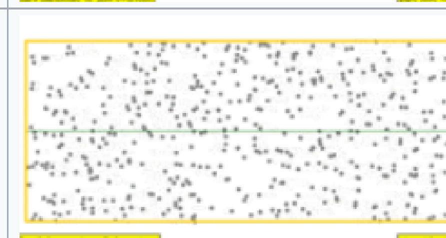
$c_k = \sqrt{a_k^2 + b_k^2}$ for $k = 1, 2, \ldots$

- The spectrum of a (periodic) signal defined to be:

  stem plot of amplitudes $\{c_k\}$ versus frequencies $\boxed{f_k = \frac{k}{T}}$

  for $k = 0, 1, 2, \ldots$

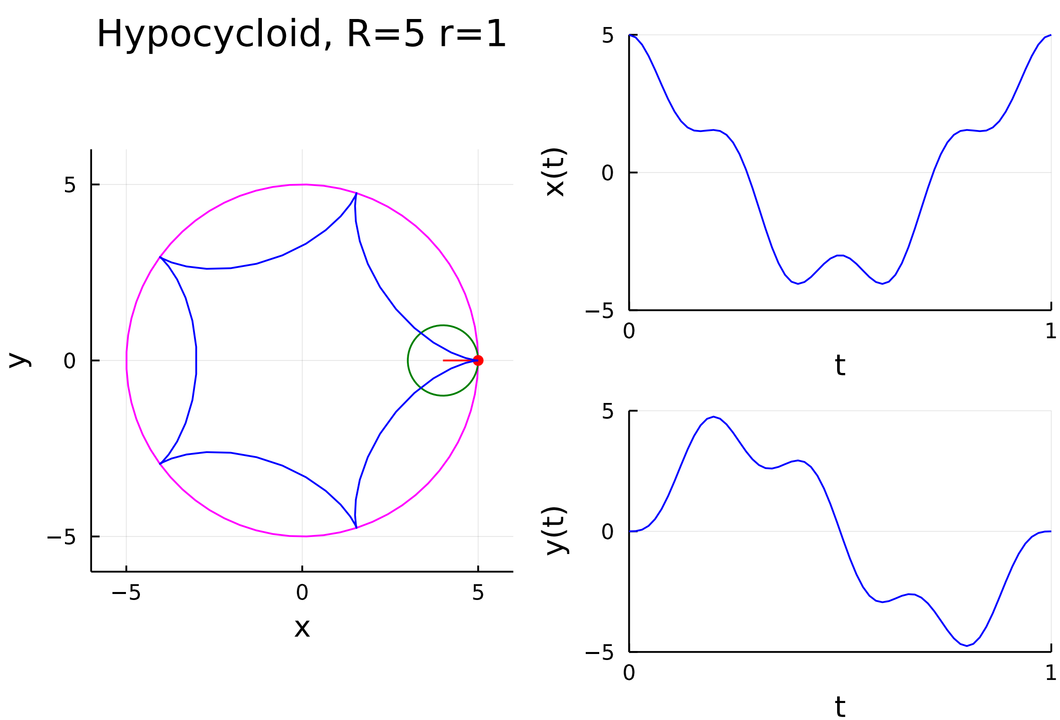
Harmonics / overtones / partials

Frequency	Order (n)	Name 1	Name 2	Name 3	Standing wave representation	Longitudinal wave representation
$1 \times f =$ 440 Hz	$n = 1$	1st partial	fundamental tone	1st harmonic		
$2 \times f =$ 880 Hz	$n = 2$	2nd partial	1st overtone	2nd harmonic		
$3 \times f =$ 1320 Hz	$n = 3$	3rd partial	2nd overtone	3rd harmonic		
$4 \times f =$ 1760 Hz	$n = 4$	4th partial	3rd overtone	4th harmonic		

[wiki]

Spectrum review: A non-music example

The following figure / demo illustrates a hypocycloid that is one special case of a spirograph.



demo: `fig_spirograph1.jl`

Formula:
$x(t) = (R-r)\cos(2\pi t) + r\cos\left(2\pi \frac{R-r}{r} t\right)$
$y(t) = (R-r)\sin(2\pi t) - r\sin\left(2\pi \frac{R-r}{r} t\right)$
Here, $R = 5$ (outer circle)
and $r = 1$ (inner circle).

Recall: a signal is any time-varying quantity...

Exercise.
Sketch spectrum of $x(t)$ (or $y(t)$). $\boxed{??}$
Is $x(t)$ band-limited? $\boxed{??}$
What is the band-limit B? $\boxed{??}$

Fourier series summary: Band-limited signals

Any signal that is

- periodic with period $= T$ and
- band-limited with maximum frequency B

can be expanded as a *finite* sum of sinusoids:

$$\boxed{x(t) = \underbrace{c_0}_{\substack{\text{DC term} \\ \text{frequency } = 0}} + \underbrace{c_1 \cos\left(2\pi \frac{1}{T} t - \theta_1\right)}_{\substack{\text{fundamental} \\ \text{frequency } = 1/T}} + \sum_{k=2}^{K} \underbrace{c_k \cos\left(2\pi \frac{k}{T} t - \theta_k\right)}_{\substack{\text{harmonics} \\ \text{frequencies } = \boxed{k/T}}}}$$

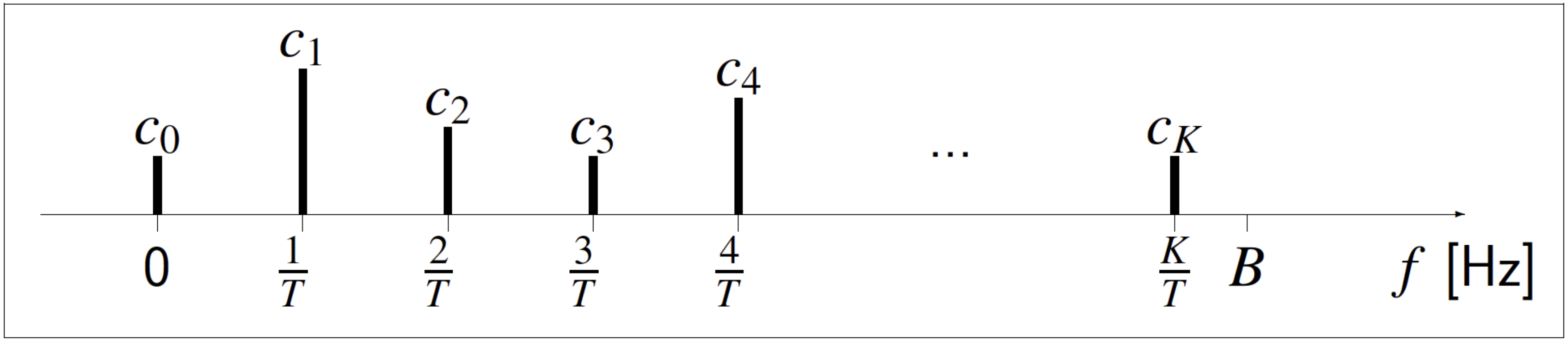
$K = \lfloor BT \rfloor$ is number of non-DC sinusoidal components. (units?)

$\boxed{2K + 1 = 2\lfloor BT \rfloor + 1}$ coefficients that characterize the signal:

$$\boxed{\{c_0, c_1, \ldots, c_K, \theta_1, \ldots, \theta_K\}}$$

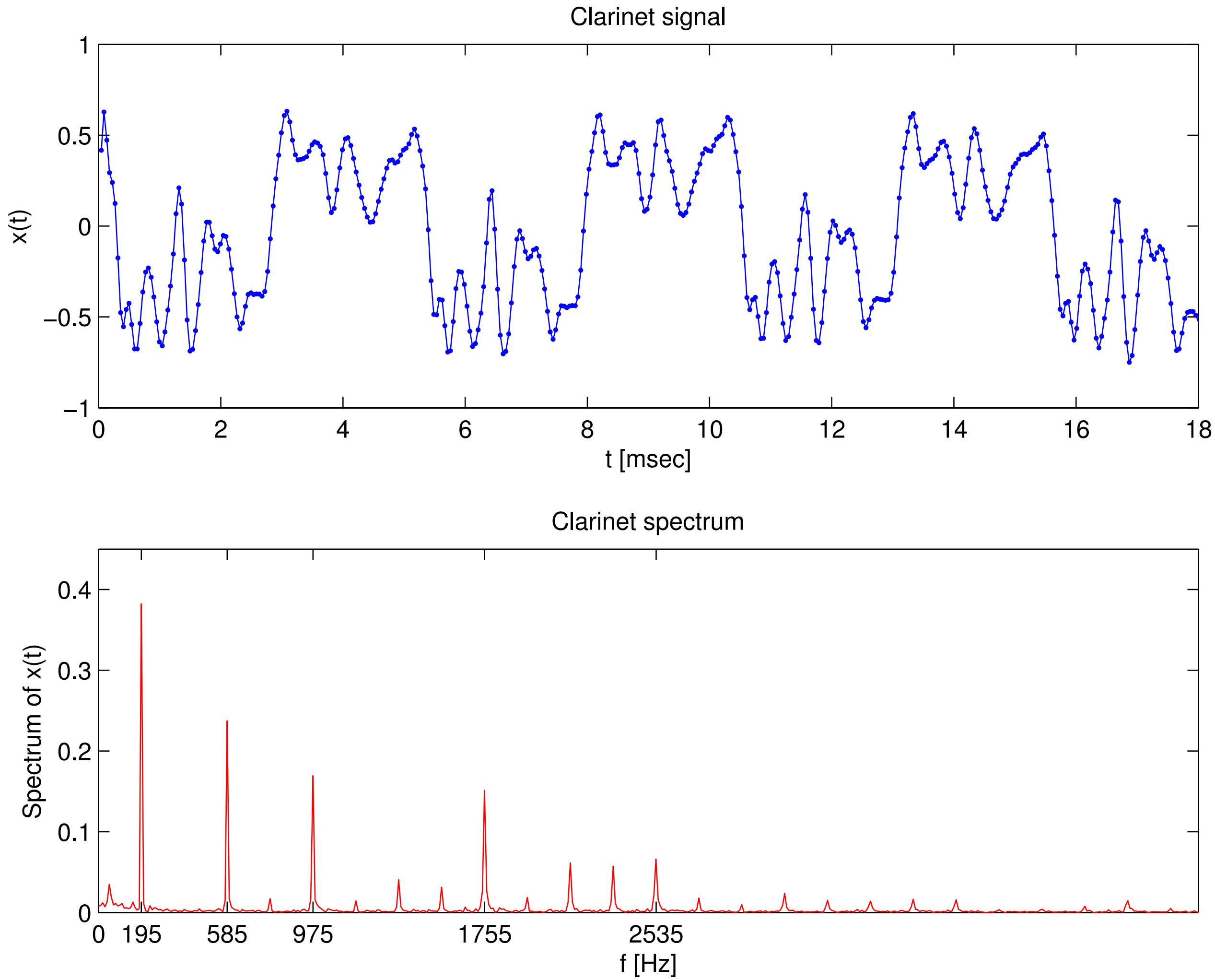
The summation formula above is precise but not "visual."
Signal spectrum: graphical display of the coefficients $\{c_k\}$.

Spectrum of a T-periodic, band-limited signal



- This is called a line spectrum.
- Periodic signals have "equally spaced" lines.
- Some (or even many) of the amplitudes c_k can be zero (no line).
- The phases $\{\theta_k\}$ are unimportant for monophonic music. (Stereo phase test.)
- The DC term (where $f = 0$) has amplitude $c_0 = 0$ for music usually.
- To keep things simple, we will usually pick $K = BT$ to be an integer, *i.e.*, assume B is an integer multiple of $1/T$.
- More generally, the highest frequency component has frequency $\frac{K}{T}$, where $K = \lfloor BT \rfloor$ and $\lfloor x \rfloor$ denotes the `floor` function.
- Example: $T = 0.01$ s and $B = 360$ Hz $\Longrightarrow K = \lfloor 3.6 \rfloor = 3$.

Example: Clarinet spectrum



First significant peak: fundamental frequency $= 1/T \approx 195$ Hz
(Perfectly) periodic signals have (perfect) line spectra.

Computing a signal's spectrum

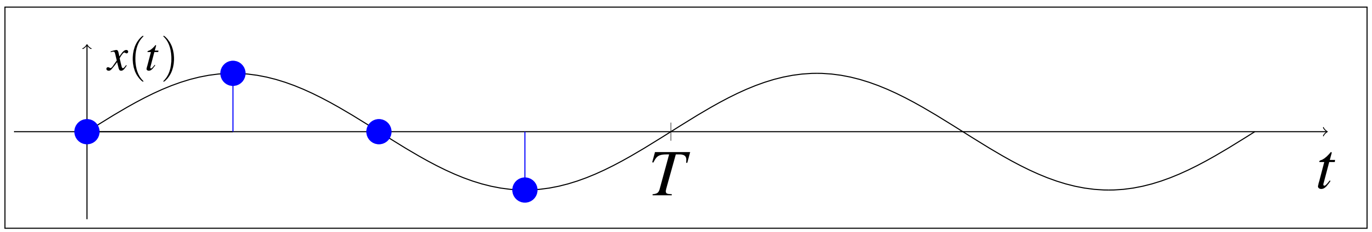
Part 4.

Sampling rate $S > 2B$

Signal sampling

Idea. To determine $2BT+1$ Fourier coefficients of a signal with period T that is band-limited to B Hz, we try taking at least $N \geq 2BT+1$ samples of the signal over one period, *e.g.*, the interval $[0,T)$.

In other words, take $N > 2BT$ samples: $x[0], x[1], \ldots, x[N-1]$.



What will be the sampling interval? $\Delta = \frac{T}{N} < \frac{T}{2BT} = \frac{1}{2B}$.

The sampling rate is: $S = \frac{\text{\# of samples}}{\text{time interval}} = \frac{N}{T} > \frac{2BT}{T} = 2B$.

Must sample *faster* than twice the maximum frequency: $\boxed{S > 2B.}$

2B or not 2B

The formula $\boxed{S > 2B}$ is one of the most important in DSP!
It is the foundation for all digital audio and video and more.
CD players use a sampling rate of 44.1 kHz. Why? ??

Where did T go?
The period T need not affect the sampling rate!
We can choose T arbitrarily large.

Amazing fact #2 (discovered by Claude Shannon 60+ years ago):

Nyquist-Shannon sampling theorem:

If we sample a band-limited signal $x(t)$ at a rate $S > 2B$, then
we can recover the signal from its samples $x[n] = x(n/S)$. (EECS 216)

Conversely:
sampling too slowly can cause bad effects called aliasing.
Example: wagon wheels in Western movies.

Claude Shannon: Father of information theory

1916-2001
Born in Petosky, raised in Gaylord, MI.
UM EE Class of 1936.
Claude Shannon's statue is outside EECS.

`http://www.computerhistory.org/collections/accession/102665758` circa 1980

cf. finite element models used by, *e.g.*, mechanical and aero engineers

Computing a signal's spectrum

Part 5.

By hand by solving systems of equations (recommended reading)

A small example: Problem statement

Given:
- $\circ$ $x(t)$ is periodic with period $T = 0.01$ second.
- $\circ$ $x(t)$ is band-limited with maximum frequency $B = 100$ Hz.

How many coefficients must we find? ??

$$x(t) = a_0 + a_1 \cos(2\pi 100 t) + b_1 \sin(2\pi 100 t)$$

What sampling rate is needed? ??

We choose $S = 400$ Hz, sample the signal, and observe:

n	1	2	3	4	$\cdots$	sample #
$t = n/S$	0.0025	0.0050	0.0075	0.0100	$\cdots$	seconds
$x[n] = x(n/S)$	40	10	20	50	$\cdots$	volts

Now we find the coefficients from these four signal samples.

A small example: Problem solution

n	1	2	3	4	sample #
$t = n/S$	0.0025	0.0050	0.0075	0.0100	seconds
$x[n] = x(n/S)$	40	10	20	50	volts

Recall: $x(t) = a_0 + a_1 \cos(2\pi 100 t) + b_1 \sin(2\pi 100 t)$.
Substituting sample times $t = n/S$ into Fourier series expansion:

$$x(n/400) = a_0 + a_1 \cos(\pi n/2) + b_1 \sin(\pi n/2).$$

$$\begin{aligned} x(1/400) &= a_0 + a_1 \cos(\pi 1/2) + b_1 \sin(\pi 1/2) = a_0 + b_1 = 40 \\ x(2/400) &= a_0 + a_1 \cos(\pi 2/2) + b_1 \sin(\pi 2/2) = a_0 - a_1 = 10 \\ x(3/400) &= a_0 + a_1 \cos(\pi 3/2) + b_1 \sin(\pi 3/2) = a_0 - b_1 = 20 \end{aligned}$$

Solve (by hand!) for 3 unknowns from 3 equations:

$$\begin{aligned} a_0 + b_1 &= 40 \\ a_0 - a_1 &= 10 \implies a_0 = 30, \quad b_1 = 10, \quad a_1 = 20 \\ a_0 - b_1 &= 20 \end{aligned}$$

Computing a signal's spectrum

Part 6.

Using general Fourier series solution

General Fourier series method

Signal $x(t)$ has period$=T$ seconds; band-limited to B Hz.
Signal is sampled at $S > 2B$ samples/second, using $t = n/S$.
Given data $x[n] = x(n/S)$ (N values from A/D converter):

$$\left[\, x\left(\tfrac{1}{S}\right),\; x\left(\tfrac{2}{S}\right),\; \cdots\; x\left(\tfrac{N}{S}\right) \,\right]$$

(sampling every $1/S$ seconds, up to N/S seconds).

Solve for $2BT+1$ unknowns from N equations.
To have a unique solution, we need $N \geq 2BT+1$.

Replace t with sample times $1/S, 2/S, \ldots, N/S$ in the equation:

$$\begin{aligned} x(t) = a_0 + a_1 \cos\left(2\pi \frac{1}{T} t\right) + \cdots + a_k \cos\left(2\pi \frac{k}{T} t\right) + \cdots + a_{BT} \cos\left(2\pi \frac{K}{T} t\right) \\ + b_1 \sin\left(2\pi \frac{1}{T} t\right) + \cdots + b_k \sin\left(2\pi \frac{k}{T} t\right) + \cdots + b_{BT} \sin\left(2\pi \frac{K}{T} t\right). \end{aligned}$$

where $K = \lfloor BT \rfloor$.
The frequency of the "kth term" in this sum is $f = \frac{k}{T}$.

General Fourier series solution

Amazing fact #3:
There is a closed-form solution (aka plug-and-chug)
to the linear system of equations.

Used by Carl Friedrich Gauss (the German "Prince of Mathematicians")
in 1805 for interpolating asteroid orbits [1]. (We call it Fourier anyway.)

$$a_0 = \frac{1}{N}\sum_{n=1}^{N} x[n] = \text{average value of } x[n]$$

$$a_k = \frac{2}{N}\sum_{n=1}^{N} x[n]\cos(2\pi nk/N),\ k = 1,\ldots,BT$$

$$b_k = \frac{2}{N}\sum_{n=1}^{N} x[n]\sin(2\pi nk/N),\ k = 1,\ldots,BT$$

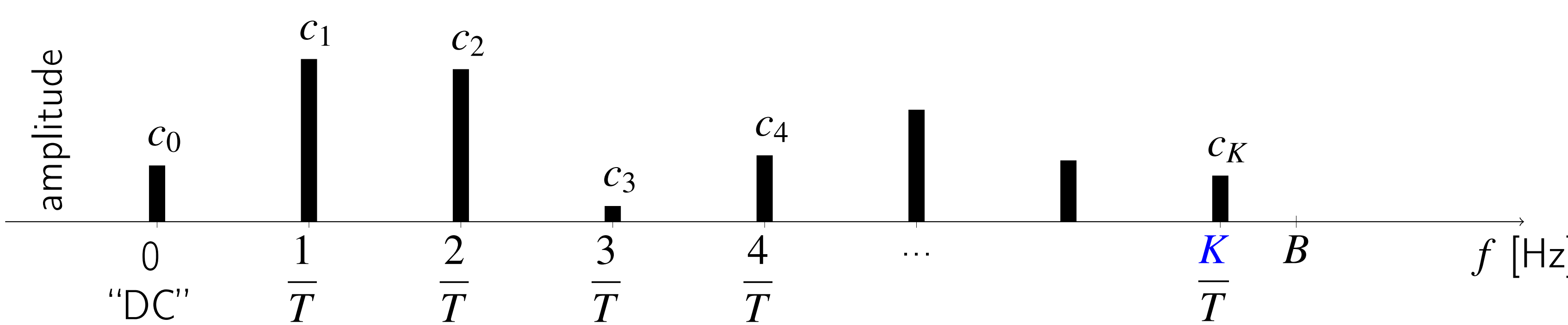
- Gives us all $2BT+1$ coefficients from N signal samples (LHS from RHS)
- Need number of samples $N \geq 2BT+1$
- Where does this solution originate? Read the Appendix of Lab 3.

Big picture for computing spectra

band-limited analog signal $x(t)$ $\rightarrow$ $\boxed{\text{Sample } S > 2B \text{ A/D converter}}$ $\rightarrow$ digital signal $x[n]$

$\rightarrow$ $\boxed{\text{Fourier series solution formula}}$ $\rightarrow$ Fourier series coefficients $\{a_k, b_k\}$

$\rightarrow$ $\boxed{c_k = \sqrt{a_k^2 + b_k^2}}$ $\rightarrow$ spectrum



A small example: Revisited (read)

Given:

n	1	2	3	4	sample #
$x[n] = x(n/S)$	40	10	20	50	volts

$T = 0.01$ sec, $S = 400\,\text{Hz}$
$B = 100\,\text{Hz}$

Find $x(t)$ using the general solution with $N = 4$:

$$a_0 = \frac{1}{N}\sum_{n=1}^{N} x[n] = \frac{1}{4}(40+10+20+50) = 30$$

$$a_1 = \frac{2}{N}\sum_{n=1}^{N} x[n]\cos\left(2\pi n\frac{1}{N}\right)$$

$$= \frac{2}{4}\left[40\cos\left(2\pi\frac{1}{4}\right) + 10\cos\left(2\pi\frac{2}{4}\right) + 20\cos\left(2\pi\frac{3}{4}\right) + 50\cos\left(2\pi\frac{4}{4}\right)\right] = 20$$

$$b_1 = \frac{2}{N}\sum_{n=1}^{N} x[n]\sin\left(2\pi n\frac{1}{N}\right)$$

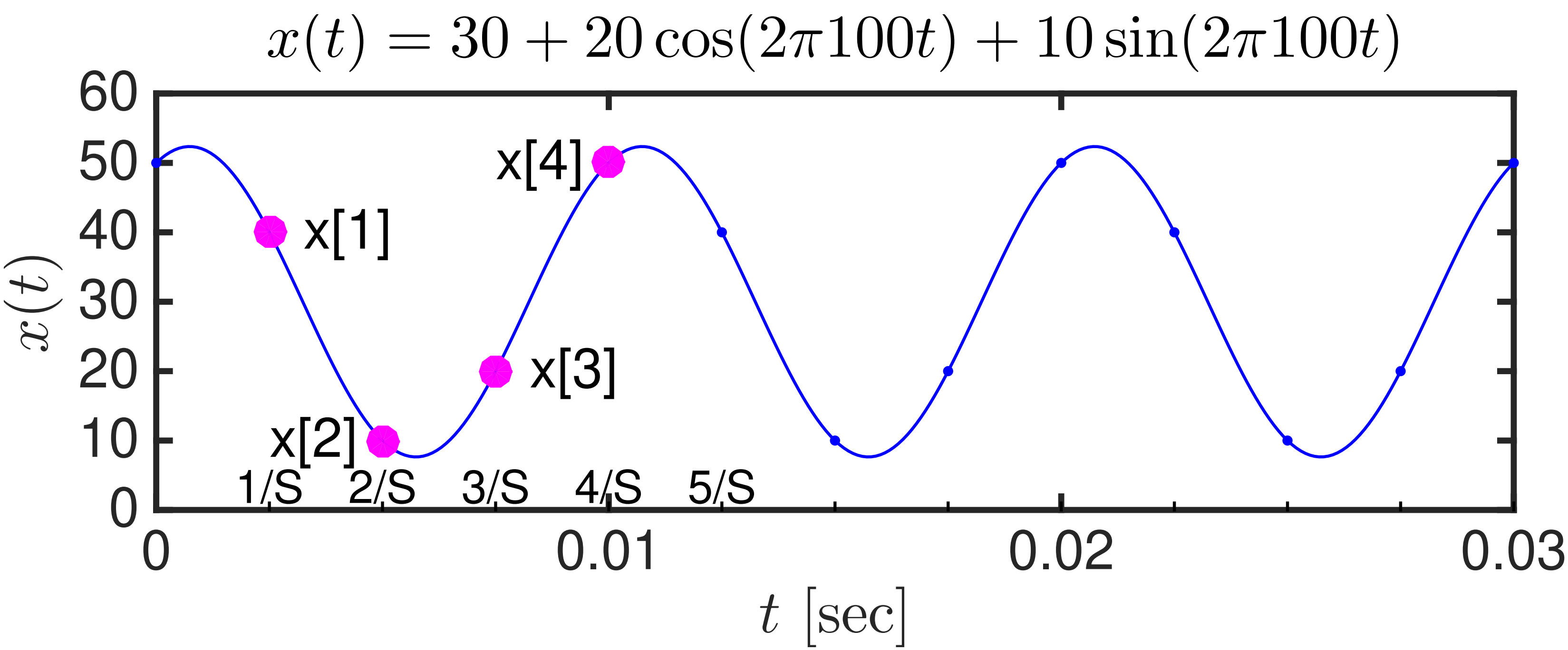
$$= \frac{2}{4}\left[40\sin\left(2\pi\frac{1}{4}\right) + 10\sin\left(2\pi\frac{2}{4}\right) + 20\sin\left(2\pi\frac{3}{4}\right) + 50\sin\left(2\pi\frac{4}{4}\right)\right] = 10$$

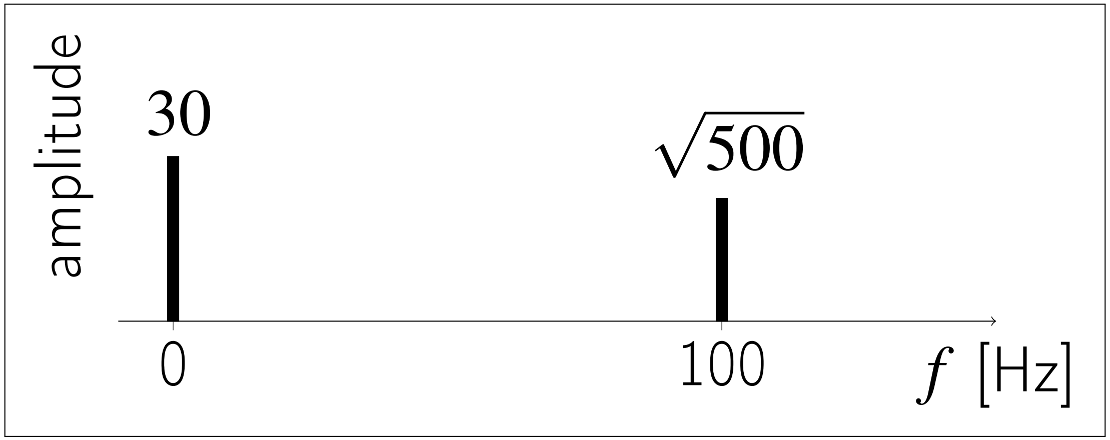
Answer: $x(t) = 30 + 20\cos(2\pi 100t) + 10\sin(2\pi 100t)$.

Check answer: What are the values of $x(2/S)$, $x(3/S)$, and $x(5/S)$?

$\boxed{??}$

Plots for small example





General solution: Remarks

- Using general Fourier series solution is much easier than solving by hand for $2BT+1$ unknowns from N simultaneous linear equations if N and $2BT$ are large!
- But calculating by hand is still slow, and so 1805...

Computing a signal's spectrum

Part 7.

Using fast Fourier transform (FFT), *e.g.*, in `Julia`

Shifting to the fast lane: FFT

If you counted how many multiplies are needed for the previous closed-form solution it would be about N^2.

Amazing fact #4:
There is a clever way to compute those summations
with only $O(N \log_2 N)$ multiplies.

Called the Fast Fourier transform (FFT). (EECS 351)

- Made famous by 1965 publication
  by James W. Cooley (1926-) and John W. Tukey (1915-2000)
- The motivating application for the Cooley–Tukey FFT was faster sensors for nuclear arms monitoring during the cold war.
- Reinvention of a method that Carl Friedrich Gauss used (by hand!) in 1805. (Theoria interpolationis methodo nova tractata) [1]

The Big 10 of algorithms

from *SIAM News*, Volume 33, Number 4

The Best of the 20th Century: Editors Name Top 10 Algorithms

By Barry A. Cipra

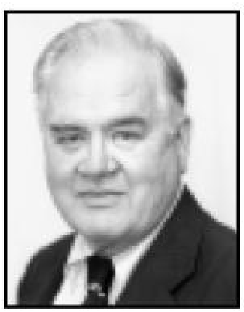
Algos is the Greek word for pain. *Algor* is Latin, to be cold. Neither is the root for *algorithm*, which stems instead from al-Khwarizmi, the name of the ninth-century Arab scholar whose book *al-jabr wa'l muqabalah* devolved into today's high school algebra textbooks. Al-Khwarizmi stressed the importance of methodical procedures for solving problems. Were he around today, he'd no doubt be impressed by the advances in his eponymous approach.

Some of the very best algorithms of the computer age are highlighted in the January/February 2000 issue of *Computing in Science & Engineering*, a joint publication of the American Institute of Physics and the IEEE Computer Society. Guest editors Jack Don-garra of the University of Tennessee and Oak Ridge National Laboratory and Fran-cis Sullivan of the Center for Comput-ing Sciences at the Institute for Defense Analyses put togeth-er a list they call the "Top Ten Algorithms of the Century."

James Cooley

1965: James Cooley of the IBM T.J. Watson Research Center and John Tukey of Princeton University and AT&T Bell Laboratories unveil the **fast Fourier transform**.

Easily the most far-reaching algo-rithm in applied mathematics, the FFT revolutionized signal processing. The underlying idea goes back to Gauss (who needed to calculate orbits of asteroids), but it was the Cooley–Tukey paper that made it clear how easily Fourier transforms can be computed. Like Quicksort, the FFT relies on a divide-and-conquer strategy to reduce an ostensibly $O(N^2)$ chore to an $O(N \log N)$ frolic. But unlike Quick- sort, the implementation is (at first sight) nonintuitive and less than straightforward. This in itself gave computer science an impetus to investigate the inherent complexity of computational problems and algorithms.



John Tukey

Using `Julia`/ FFT to compute spectra

Given a 1D array `x` of N samples of a signal $x(t)$:

$$\texttt{x} = \left[\, x\!\left(\tfrac{0}{S}\right),\; x\!\left(\tfrac{1}{S}\right),\; \cdots\; x\!\left(\tfrac{N-1}{S}\right) \,\right]$$

`Julia`'s `F=fft(x)` computes the following 1D array of N values:

$$\texttt{F[k+1]} = \sum_{n=0}^{N-1} x\!\left(\frac{n}{S}\right) \mathrm{e}^{-\imath 2\pi nk/N}, \quad k = 0, \ldots, N-1,$$

(with $O(N \log_2 N)$ multiplies!) where Euler's identity is $\mathrm{e}^{\imath\theta} = \cos(\theta) + i \sin(\theta)$, $i = \sqrt{-1}$.

What you need to know:

$a_0 =$ `mean(x)` $=$ 1st element of `(1/N)*fft(x)`

$a_k =$ element `k+1` of `(2/N)*real(fft(x))`, $k = 1, 2, \ldots, N/2$

$b_k =$ element `k+1` of `(-2/N)*imag(fft(x))`, $k = 1, 2, \ldots, N/2$

$c_k =$ element `k+1` of `(2/N)*abs.(fft(x))`, $k = 1, 2, \ldots, N/2$.

We need to ensure $S > 2B$ (maximum frequency of $x(t)$) and $N \geq 2BT + 1$.

Caution: `Julia` array index 1 to N, but math coefficient index starts at 0. Note "$k+1$."

Big picture for computing spectra via FFT

band-limited analog signal $x(t)$ → **[** Sample $S > 2B$ A/D converter **]** → digital signal $x[n]$

Julia 1D array `x` → **[** FFT `fft(x)` **]** → FFT coefficients

→ **[** $c_k = \sqrt{a_k^2 + b_k^2}$ use: `(2/N)*abs.(fft(x))` **]** → spectrum

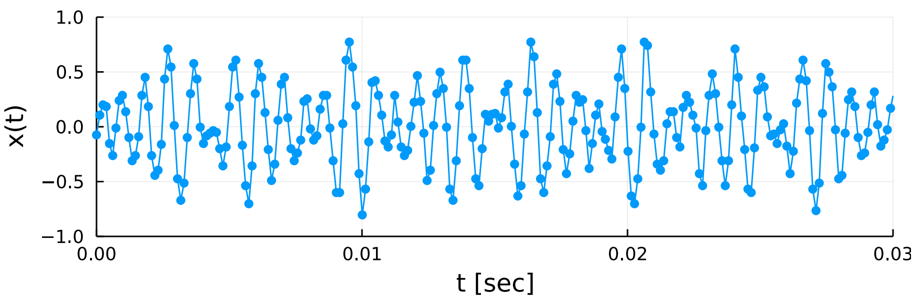
→ **[** visualize use stem plot **]**

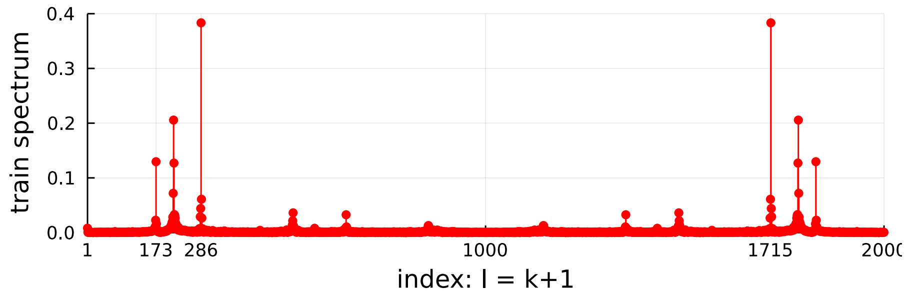
Example - rough spectrum plot

```
# fig_train_spectrum1a.jl
using WAV: wavread
using FFTW: fft
using Plots: plot, default; default(markerstrokecolor=:auto, label="")

(y, S) = wavread("../synth/train-whistle.wav")
N = 2000; n = 0:N-1; t = n/S
x = y[8000 .+ n] # just 2000 samples of it

p1 = plot(t, x, marker=:circle, xlim=(0,0.03), ylim=(-1,1), xlabel = "t [sec]", ylabel = "x(t)")

p2 = plot(2/N*abs.(fft(x)), line=:stem, marker=:circle, color=:red,
   xlabel = "index: l = k+1", ylabel = "train spectrum", xlim = (1,N), ylim = (0, 0.4),
   xtick = [1, 173, 286, N/2, N+1-286, N]) #ytick = [0, 0.5],

plot(p1, p2, layout = (2,1)) #; savefig("fig_train_spectrum1a.pdf")
```

Q0.1 Periodic?

A: True B: False ??





Mirror symmetry of `fft` output

`fft` output has some even/odd symmetry due to $\mathrm{e}^{-\imath 2\pi nk/N}$ term.
For a real array `x` of length N in `Julia`:

`(2/N)*real(fft(x))` gives

$$[2a_0 \; \underbrace{a_1 \; a_2 \; \ldots \; a_{N/2-2} \; a_{N/2-1}} \; a_{N/2} \; \underbrace{a_{N/2-1} \; a_{N/2-2} \; \ldots \; a_2 \; a_1}]$$

`(-2/N)*imag(fft(x))` gives

$$[0 \; \underbrace{b_1 \; b_2 \; \ldots \; b_{N/2-2} \; b_{N/2-1}} \; 0 \; \underbrace{-b_{N/2-1} \; -b_{N/2-2} \; \ldots \; -b_2 \; -b_1}]$$

`(2/N)*abs.(fft(x))` gives

$$[2c_0 \; \underbrace{c_1 \; c_2 \; \ldots \; c_{N/2-2} \; c_{N/2-1}} \; c_{N/2} \; \underbrace{c_{N/2-1} \; c_{N/2-2} \; \ldots \; c_2 \; c_1}]$$

`angle.(fft(x))` gives

$$[0(\text{or } \pi) \; \underbrace{-\theta_1 \; \ldots \; -\theta_{N/2-1}} \; 0(\text{or } \pi) \; \underbrace{\theta_{N/2-1} \; \ldots \; \theta_2 \; \theta_1}]$$

For plotting spectra, ignore the redundant 2nd half of the `fft` output array!

A small example: `Julia`/ FFT approach (read)

Given:

n	1	2	3	4	sample #
$x[n] = x(n/S)$	40	10	20	50	volts

$T = 0.01$ sec, $S = 400$ Hz

Recall that for $N = 4$, the FFT needs this array of samples:

$$\texttt{x} = \left[\, x\!\left(\frac{0}{S}\right) \;\; x\!\left(\frac{1}{S}\right) \;\; x\!\left(\frac{2}{S}\right) \;\; x\!\left(\frac{3}{S}\right) \,\right]$$

Wait, no $x(0)$ in our table!?
But because $x(t)$ is T-periodic: $x\!\left(\frac{N}{S}\right) = x\!\left(\frac{ST}{S}\right) = x(T) = x(0)$.
Here $x(0) = x(4/S) = 50$.

```
x = [50, 40, 10, 20]
2/4 * real(fft(x))
```

gives: `[60 20 0 20]` *cf.* $[2a_0 \; a_1 \; a_2 \; a_1] \Longrightarrow a_0 = 60/2 = 30$ and $a_1 = 20$

```
-2/4 * imag(fft(x))
```

gives: `[0 10 0 -10]` *cf.* $[0 \; b_1 \; 0 \; -b_1] \Longrightarrow b_1 = 10$

Answer: $x(t) = 30 + 20\cos(2\pi 100 t) + 10\sin(2\pi 100 t)$.

Using `fft` is the easiest way of all to find Fourier series coefficients!

Frequencies for `Julia`/FFT spectra

The frequency of the kth term in a Fourier series is

$$\boxed{f = \frac{k}{T}.}$$

This formula is not so helpful if the period T is unknown!
In music transcription or pitch tracking, the pitch is unknown,
so the (fundamental) period is unknown.

Fact. When applying the FFT to N samples of signal using sampling rate S, the frequency of the kth term is:

$$\boxed{f = \frac{k}{N} S,} \quad k = 0, \ldots, N/2.$$

Caution. The kth term appears as `Julia` index $\boxed{\texttt{l} = k + 1.}$

array	$[2c_0$	c_1	c_2	$\ldots$	$c_{N/2-1}$	$c_{N/2}$	$c_{N/2-1}$	$\ldots$	c_2	$c_1]$
index	1	2	3	$\ldots$	$N/2$	$N/2+1$	$N/2+2$	$\ldots$	$N-1$	N

Example. Spectrum of unknown signal

We have $N = 10$ samples of a signal at $S = 500$ Hz stored in `x`.

`(2/N)*real(fft(x))` gives `[0 0 0 7 0 0 0 7 0 0]`

`(-2/N)*imag(fft(x))` gives `[0 0 0 0 0 0 0 0 0 0]`

What is the signal? Recall: $[2a_0 \; \underbrace{a_1 \; a_2 \; a_3} \; a_4 \; \underbrace{a_3 \; a_2 \; \dots \; a_1}]$

Here $a_3 = 7$. Caution: 4th element of array is 7.
`Julia` index `l=4` corresponds to frequency term $k = 3$.
Using $f = \frac{k}{N} S = \frac{3}{10} 500 = 150$:

$$x(t) = 7 \cos(2\pi 150 t)$$

which is a 150Hz sinusoidal signal.
Ignore the other `7` in array index $N + 1 - k = 10 + 1 - 3 = 8$.

Spectrum and FFT of a single sinusoid

Example. $x(t) = 7\cos(2\pi 440 t)$

What is its spectrum? ??

What happens if we use `Julia`'s `fft` to find its spectrum?

```
using FFTW: fft
using Plots: plot
N = 2000; t = (0:N-1)/1000
x = 7 * cos.(2π * 440 * t) # type \pi<tab>
plot((2/N) * abs.(fft(x)), line=:stem)
```

What will the stem plot look like?

Hint. Determine S and N and use $f = S\frac{k}{N}$.

$\boxed{??}$

Q0.2 Enter one of the crucial values along the horizontal axis.

$\boxed{??}$

The two figures are closely related, but differ.

Using `Julia`/ FFT to compute spectra

Given a vector `x` of N samples of a signal $x(t)$ taken at rate S:

$$\mathtt{x} = \left[\, x\!\left(\tfrac{0}{S}\right),\ x\!\left(\tfrac{1}{S}\right),\ \cdots\ x\!\left(\tfrac{N-1}{S}\right) \,\right],$$

`Julia`'s `fft` can compute amplitudes $\{c_k\}$ and phases $\{\theta_k\}$:

`(2/N)*abs.(fft(x))` gives

$$[2c_0\ \underbrace{c_1\ c_2\ \ldots\ c_{N/2-2}\ c_{N/2-1}}\ c_{N/2}\ \underbrace{c_{N/2-1}\ c_{N/2-2}\ \ldots\ c_2\ c_1}_{\text{mirror image}}]$$

`angle.(fft(x))` gives

$$[0(\text{or } \pi)\ \underbrace{-\theta_1\ \ldots\ -\theta_{N/2-1}}\ 0(\text{or } \pi)\ \underbrace{\theta_{N/2-1}\ \ldots\ \theta_2\ \theta_1}_{\text{mirror image}}]$$

In other words (note "index $= k+1$"):

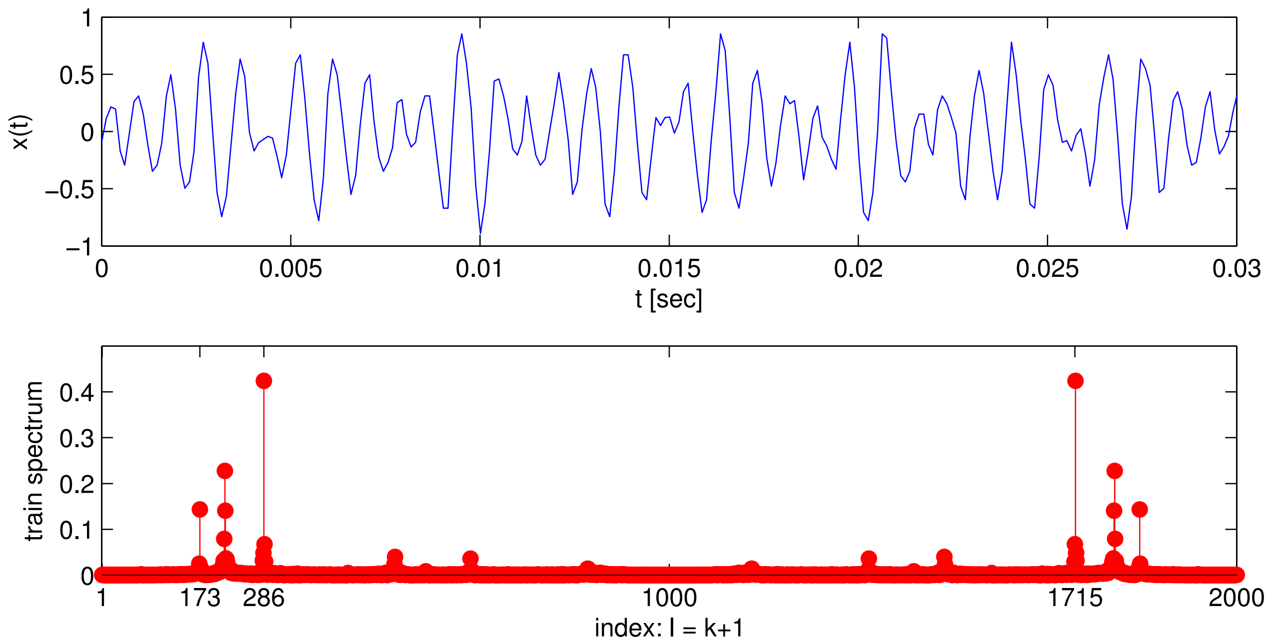
$c_0 =$ `mean(x)` $=$ 1st element of `1/N*(fft(x))`

$c_k = (k+1)$th element of `2/N*abs.(fft(x))` , $k = 1, 3, \ldots, N/2$

frequency of kth term: $\boxed{f = \frac{k}{N} S.}$

Spectrum of real signal #1

See `Julia` code on p. 29.
Full $x(t)$. play
2000-point snippet $x[t]$. play



3 big peaks in spectrum at `Julia` indexes `173 217 286`

Fourier series approximation of train signal

```
Xf = fft(x)
k1 = [173, 217, 286]
2/N*abs.(Xf[k1])
```

resulting amplitudes (c_k): `0.13 0.21 0.38`

```
angle.(Xf[k1])
```

resulting phases ($-\theta_k$ in radians): `1.75, -1.49, -0.66`

```
(k1 .- 1) / N * S
```

resulting frequencies (f_k in Hz): `705, 885, 1167`

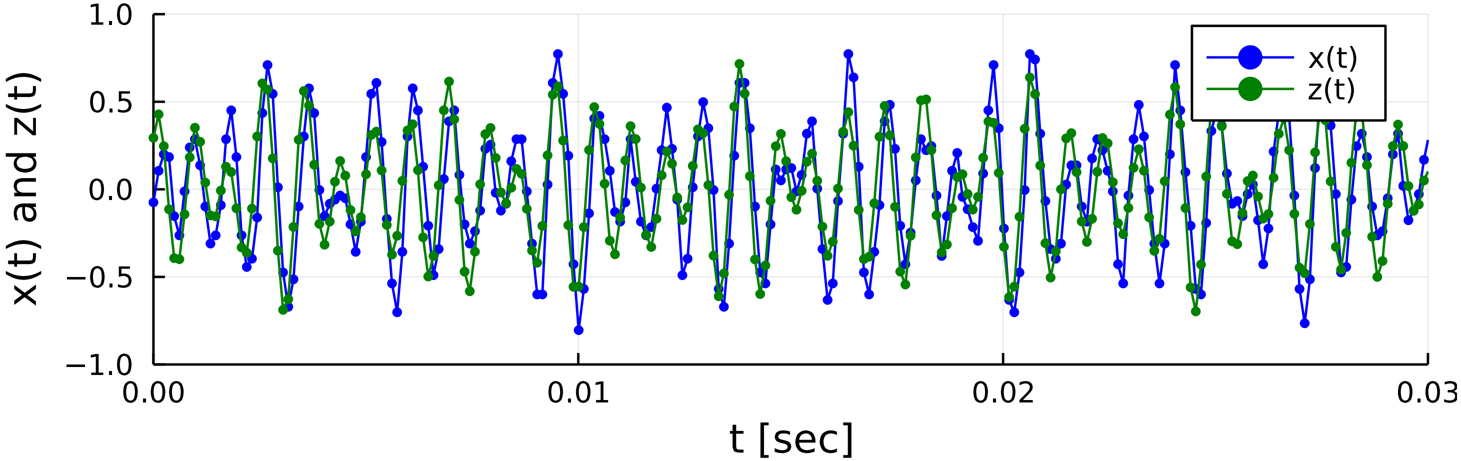
Thus a 3-term Fourier series approximation is $x(t) \approx z(t)$ where
$z(t) = 0.13\cos(2\pi 705 t + 1.75) + 0.21\cos(2\pi 885 t - 1.49) + 0.38\cos(2\pi 1167 t - 0.66)$.

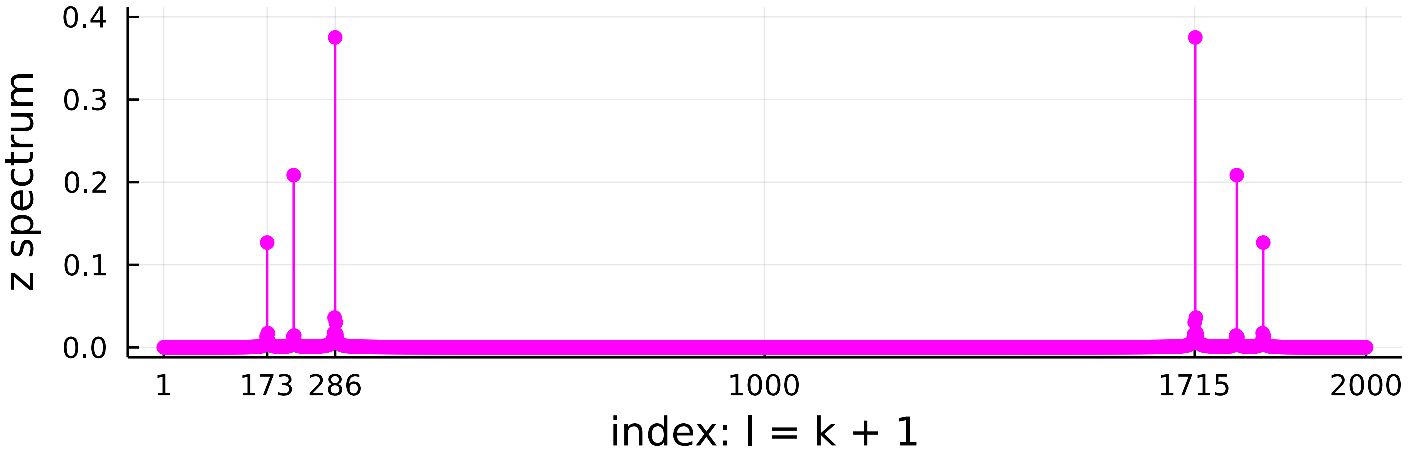
Aside: 705, 885, 1167 Hz roughly correspond notes F5 A5 D6

Aside for musicians: What chord? $\boxed{??}$

Spectrum of synthesized train whistle signal

```
z = 0.13 * cos.(2π*705*t .+ 1.75) + 0.21 * cos.(2π*885*t .- 1.49) +
    0.38 * cos.(2π*1167*t .- 0.66)
```





play

Mission accomplished

We just found the frequencies of 3 notes in a chord
for a real-world music signal that was:

- *not* a pure sinusoid,
- consisted of multiple notes (a chord),
- and was recorded with a microphone so contains noise.

The FFT is a key tool for music transcription by computers.

Required more complicated math than previous $\arccos$ method.
But the final FFT method is very practical.

MP3 encoders, pitch trackers, tempo converters, vocal auto-tune, ...
all use FFT-type methods based on a signal's spectrum.

Spectrum of a live recorded signal

Basic audio recording with `Julia`:

```
using Sound: record, sound
using Plots: plot
using FFTW: fft
record(0.001) # warm-up
println("begin")
x, S = record(5)
x = x[1:2:end]; S ÷= 2 # reduce memory
Nx = length(x)
t = (1:Nx)/S
p0 = plot(t, x, xlabel="t [s]", ylabel="x(t)")
```

This records `5` seconds of monaural audio sampled at `8000` Hz and stores the results in vector `x`. Requires a microphone.

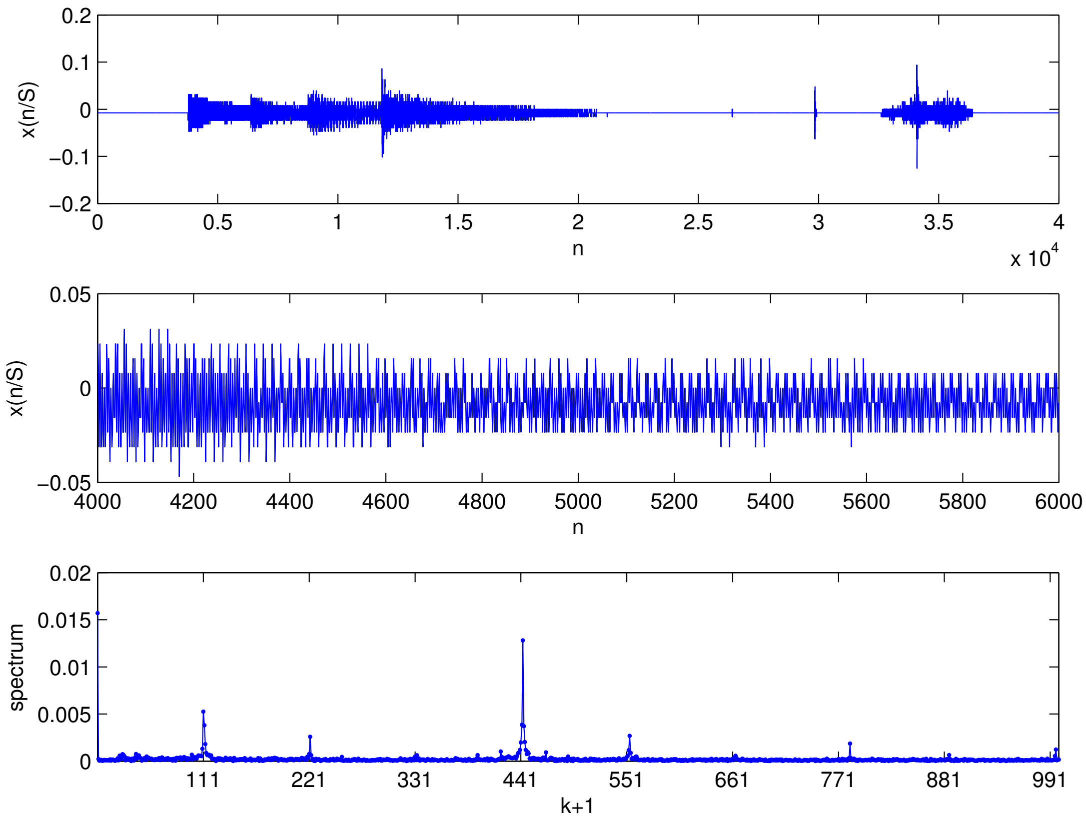
Key formula: $\boxed{f = \frac{k}{N} S}$ [wiki]

(guitar "transcription" demo here using `fig/l05/record2-fft1.jl` `fig/l05/record2-fft2.jl`)

Plotting code

```
using Plots: plot!, default; default(label="")
n = Int[2.0 * S] .+ (1:2000)
p1 = deepcopy(p0)
plot!(p1, t[n], x[n], color=:magenta)
y = x[n]; Ny = length(y) # segment
p2 = plot(y, xlabel="n (sampling rate $S Hz)", ylabel="y[n]")
lmax = 81
p3 = plot(2/Ny * abs.(fft(y)), line=:stem,
    title="spectrum of y (zoomed), fmax=$((lmax-1)/Ny*S) Hz",
    xlabel="frequency index l=k+1", xlims=(0,lmax))
plot(p1, p2, p3, layout=(3,1))
```

DSP, FFT and "Stairway to heaven"



play

play

Fundamental frequency of first note: $f = \frac{k}{N}S = \frac{110}{2000}8000 = 440\text{Hz}.$

Q0.3 What is the lowest (nonzero) frequency (in Hz) we could find here?

??

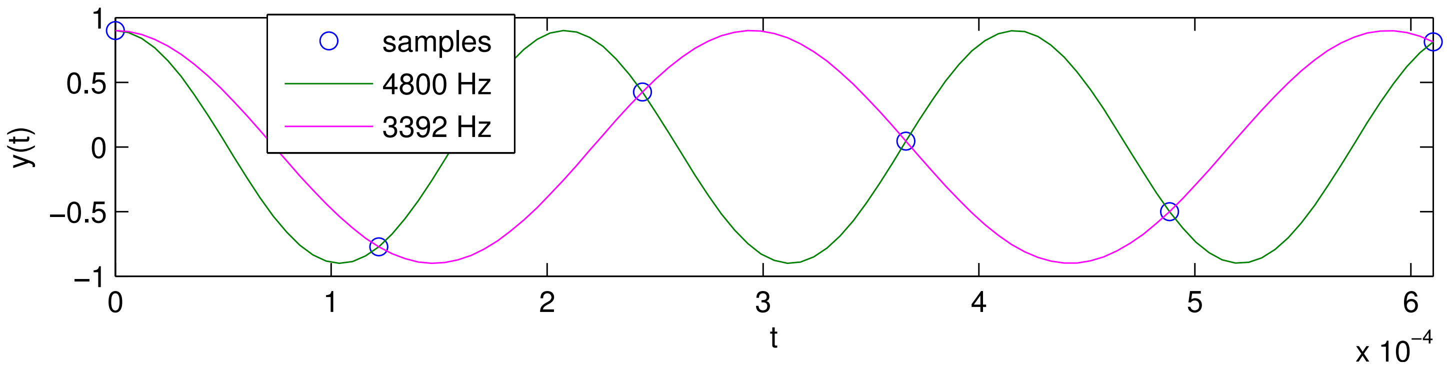
Summary so far

- Periodic signals (including musical notes) can be expanded in a Fourier series.
- The signal is a sum of sinusoids at frequencies that are integer multiples of $1/T$
- If signal is both band-limited and periodic,
  - the Fourier series has only a finite # of terms
  - we can compute coefficients from its samples if $\boxed{S > 2B}$
- There is a closed-form solution to the resulting linear system;
- `Julia` can compute it easily using `fft`.
- Use $\boxed{f = \frac{k}{N} S}$ to get frequency, where $k =$ `l` $- 1$ (`l` $=$ array index).

Aliasing: audio example

```
S = 8192; t = 0:1/S:0.3
x = 0.9*[cos.(2pi*2800*t); cos.(2pi*3800*t)]
```
play

```
y = 0.9*[cos.(2pi*3800*t); cos.(2pi*4800*t)]
```
play



arccos method says 3392 Hz, not 4800 Hz for last part of this example

Q0.4 Is $S > 2B$ here?

A: Yes B: No ??

References

[1] M. T. Heideman, D. H. Johnson, and C. S. Burrus. Gauss and the history of the fast Fourier transform. *Archive for History of Exact Sciences*, 34(3):265–77, September 1985.