$f(t)$ using the fact $\boxed{\Delta \omega = \frac{\pi}{L}}$

$$\frac{1}{2L} = \frac{\Delta\omega}{2\pi}, \text{ we get}$$

$$f(t) = \sum_{K=-\infty}^{\infty} \left[\frac{\Delta \omega}{2\pi} \int_{-L}^{L} f(u)\, e^{-i\omega_K u}\, du \right] e^{i\omega_K t}$$

$$= \sum_{K=-\infty}^{\infty} \frac{\Delta\omega}{2\pi} \int_{-L}^{L} f(u)\, e^{-i\omega_K u} \cdot e^{i\omega_K t}\, du.$$

As $L \to \infty$

$$= \lim_{L\to\infty} \sum_{K=-\infty}^{\infty} \frac{\Delta\omega}{2\pi} \int_{-L}^{L} f(u)\, e^{-i\omega_K u} \cdot e^{i\omega_K t}\, du$$

$$= \int_{-\infty}^{\infty} \frac{d\omega}{2\pi} \int_{-\infty}^{\infty} f(u)\, e^{-i\omega u}\, e^{i\omega t}\, du$$

$$= \int_{-\infty}^{\infty} \frac{1}{2\pi} \int_{-\infty}^{\infty} f(u)\, e^{-i\omega u}\, e^{i\omega t}\, du\, d\omega$$

$$= \int_{-\infty}^{\infty} \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} f(u)\, e^{-i\omega u}\, du \right) e^{i\omega t}\, d\omega)$$

Define

$$\hat{f}(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(u)\, e^{-i\omega u}\, du = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt$$

Then it follows that

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega)\, e^{i\omega t}\, d\omega.$$

In practice, the above Fourier transform is often a straight point for analysis of signals. However, discrete series are sufficient approximations in many cases. You'll likely encounter a Fourier transform in other courses. See for example 18.103, 18.303, 18.311, 18.354, 18.353.

$$\lim_{n\to\infty} \sum_{i=1}^{n} f(x_i)\, \Delta x = \int_a^b f(x)\, dx. \quad \text{[Riemann sums]}$$

(Riemann sums as definite integrals)

Eg: matlab

1) calculate $|c_n|$'s using the complex representation of the F.S either even or odd extension for the audio signal.

2) calc the F.T $\hat{f}(k)$

3) plot the results from (1) and (2) on the same graph & compare.

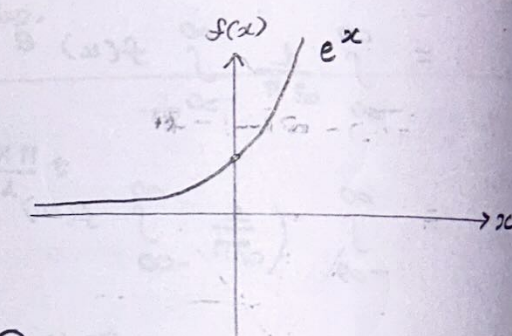
Even & odd functions

Every function $g(x)$ can be written as the sum of odd and even function

$$g(x) = g_{even}(x) + g_{odd}(x)$$

where $g_{even}(-x) = g_{even}(x)$ and $g_{odd}(-x) = -g_{odd}(x)$

Find even & odd Functions of e^x:



$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$$

$$g(x) = g_e(x) + g_o(x) \longrightarrow ①$$

$$g(-x) = g_e(-x) + g_o(-x)$$

$$\boxed{g(-x) = g_e(x) - g_o(x)} \longrightarrow ②$$

Adding ① and ②

$$g(x) + g(-x) = 2g_e(x)$$

$$g_e(x) = \frac{1}{2}\left[g(x) + g(-x)\right]$$

$$g(x) - g(-x) = 0 + g_o(x) - (-g_o(x))$$

$$g(x) - g(-x) = 2g_o(x)$$

$$g_o(x) = \frac{1}{2}\left[g(x) - g(-x)\right]$$

Manipulating series

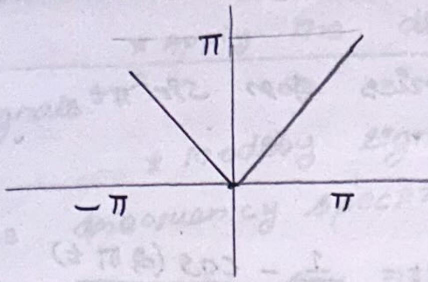
Use the fact that the Δ^{le} wave of period 2π defined by

$$g(u) = |u| \qquad -\pi \le u \le \pi \qquad \text{has}$$

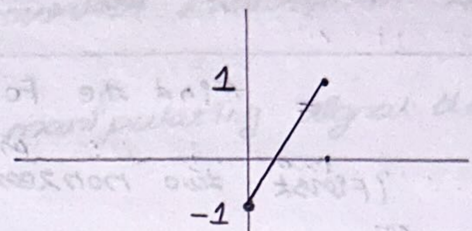
$$\text{F.S} \quad g(u) = \pi/2 - \pi/4 \left(\cos u + \frac{\cos u}{3^2} + \frac{\cos 5u}{5^2} + \dots\right)$$

To find the F.S of the even function of period 2 defined on the interval $0 \le t \le 1$ by

$$\boxed{f(t) = 2t - 1 \qquad 0 \le t \le 1}$$



$g(u) = |u|$



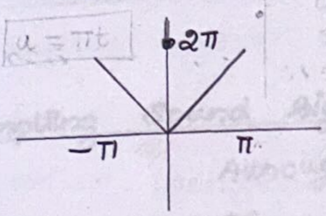
$f(t) = 2t - 1$

(Shifting by 1 upwards)

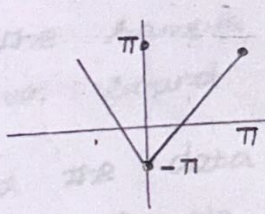
Shifting by 1 downwards

$g(u) = 2|u| \rightarrow$ scaled

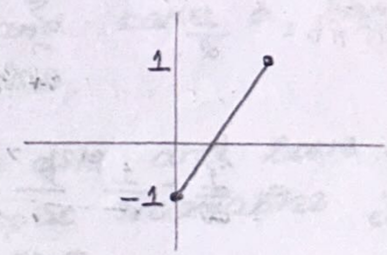
$\boxed{u = \pi t}$



$g(u) = 2|u| - \pi$



(compare)



$f(t) = 2t - 1.$

$$\boxed{\begin{aligned} g(\pi) &= 2\pi - \pi \\ &= \pi \end{aligned}}$$

$u = 0, \quad t = 0$

$u = \pi, \quad t = 1$

$\boxed{u = \pi t}$

$$f(t) = \frac{g(\pi t)}{\pi} = \frac{2g(\pi t) - \pi}{\pi}$$

$$2g(\pi t) = \frac{\pi}{2} - \frac{8}{\pi}\left(\cos u + \frac{\cos 3u}{9} + \frac{\cos 5u}{25} + \cdots\right)$$

$$2g(\pi t) = \pi - \frac{8}{\pi}\left(\cos \pi t + \frac{\cos 3\pi t}{9} + \frac{\cos 5\pi t}{25} + \cdots\right) - \pi$$

$$\frac{2g(\pi t) - \pi}{\pi} = -\frac{8}{\pi^2}\left(\cos \pi t + \frac{\cos 3\pi t}{9} + \frac{\cos 5\pi t}{25} + \cdots\right)$$

$$f(t) = -\frac{8}{\pi^2}\left(\cos \pi t + \frac{\cos 3\pi t}{9} + \frac{\cos 5\pi t}{25} + \cdots\right)$$

$$f(t) = \frac{2|u| - \pi}{\pi} = \frac{2g(\pi t) - \pi}{\pi}$$

$$f(t) = -\frac{1}{2}\left(\cos\pi t + \frac{\cos 3\pi t}{9} + \dots + \frac{\cos n\pi t}{9}\right)$$

First three non-zero term

$$f(t) = -\frac{8}{\pi^2}\left(\cos\pi t + \frac{\cos 3\pi t}{9} + \frac{\cos 5\pi t}{25}\right)$$

Find the Fourier series for $\sin^2\pi t$

(First two nonzero terms)

solu:

$$\boxed{\begin{aligned}\sin^2 x &= \frac{1-\cos 2x}{2}\\ &= \frac{1}{2} - \frac{\cos 2x}{2}\end{aligned}}$$

$$\sin^2\pi t = \frac{1}{2} - \frac{\cos(2\pi t)}{2}$$

Complex co.efficients:

$$\cos t = \frac{1}{2}\left(e^{it} + e^{-it}\right)$$

$$(\sin t = \frac{i}{2}\left(e^{-it} - e^{it}\right)$$

$$\sin^2\pi t = \frac{1}{2} - \sum_{\substack{n\\ \text{even}}} \frac{(\cos n\pi t)}{n}$$

$$= \frac{1}{2} - \frac{1}{2}\sum_{\substack{n\\ \text{even}}} \frac{1}{2}\left(e^{in\pi t} + e^{-in\pi t}\right)$$

$$= \frac{1}{2} - \frac{1}{2}\sum_{\substack{n\\ \text{even}}} \frac{1}{n}\left(e^{in\pi t} + e^{-in\pi t}\right)$$

$$c_n = \frac{1}{n}(a_n - b_n) \quad x - \frac{1}{2}$$

$$\boxed{a_n - b_n = 1}$$

The co.eff $e^{in\pi t} = 1$.

$$\therefore \quad c_0 = \frac{a_0}{2}$$

$$\boxed{c_0 = \frac{1}{2}}$$

$$\left.\begin{aligned} c_1 &= 0\\ c_{-1} &= 0\end{aligned}\right\}\text{ odd terms.}$$

$$c_{-2} = -\frac{1}{2}\left(\frac{1}{n}\right)_{n=2}$$

$$= -\frac{1}{2}\left(\frac{1}{2}\right) = -\frac{1}{4}.$$

$\therefore \cos$ is a even function

$$c_2 = -\frac{1}{2}\left(\frac{1}{2}\right)_{n=2}$$

$$= -\frac{1}{4}.$$

Discrete Fourier transform & signal processing

1) How signals are sampled in real world, and the mathematics that allows us to determine the frequencies from sampled data.

* Apply the discrete Fourier transform to real signals.
* Modify signals by manipulating signal data in the frequency spectrum.
* Identify notes & harmonics in frequency spectrum from musical audio files.
* Read frequency information from spectrograms of audio files that change in time.

Analyzing audio files

Sampling sound signals:

Although sound files sound continuous, in digital file formats, the data is being sampled discretely. In this problem, you'll learn how to read audio files, find the sample plot, plot them, and take a sample of your sound signal to analyze.

To read the data from the audio file and save the data and sample rate as the output variables y and Fs respectively, use the command.

```
[y, Fs] = audioread('audio filename.wav') → quotations must.
% y - Data from your file
% Fs - sample rate
% we will use .wav files in this problem.
% But this works for any standard audio file type.
```

The thing about sound signals is that they change in time. When we want to analyze a piece of sound signal, we typically need to trim the beginning and end, and crop our data to find some representative sample in the middle which we can analyze (using fourier transform & analyze methods.). (Because sound signal, we are working with here is one tone, we only need to trim the beginning & end where there is no sound. However, in general, determining the window of what to sample is itself part of the challenge.)

Suppose, I want to clip of the first 10% of the sound signal, and then sample 30% of the signal. The way to do this would be the following.

```
L = length(y); % Find the length of the audio signal
y= y(round (L*0.1):round (L*0.4)); % pick the range
from 10% to 40% range.
% use the function round(), floor(), or ceil(),
to guarantee that your range is given by integers.
```

our data file '1803_musicdata_guitar1.wav'

Need to save in current directory: 'save using Matlab drive'

```
[y, FS]= audioread ('1803_musicdata_guitar1.wav');
```

% we can also use sound(y, FS), But we didn't used it

%. Because, the sound signals have 2 channels, we are analyzing the first channel alone.

```
y = y(:,1);
figure (1)
plot (y)
set (gca, 'font size', 18)
L= length(y);
```

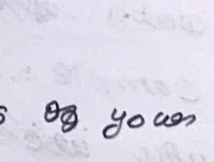
gca → current axis size, color

% Determine starting & ending points of your clipping by looking at plot 1.

```
index1 = round (L*0.1);
index2 = round (L*0.4);
newsignal = y(index1:index2);
figure (2)
plot (newsignal)
set (gca, 'fontsize', 18)
```

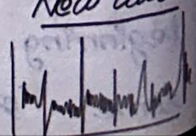
figure ↳ means

'Tabulation'



(Audio)

New audio



Discrete Fourier Transform

Review:

we can think of the Fourier series as a method that takes "any" periodic function, and replaces it with a discrete set of co-efficients ("infinitely many", but countably infinite").

$$f(t) = f(t+2L) = \sum_{k=-\infty}^{\infty} c_k \, e^{ik\pi t/L}$$

$$\boxed{c_k = \frac{1}{2L}\int_{-L}^{L} f(t)\, e^{-ik\pi t/L}\, dt}$$

The Fourier transform takes "any" function, and replaces it with a continuous function.

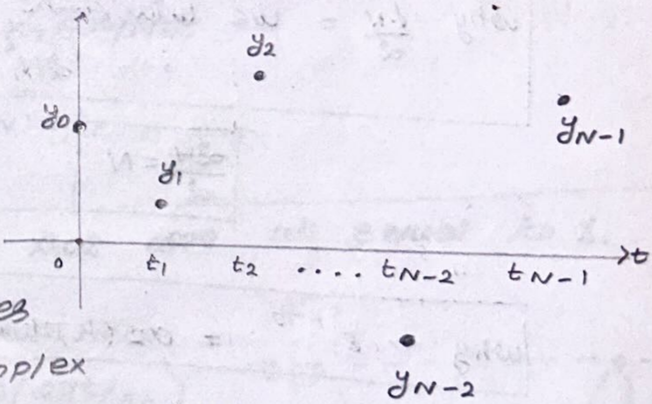
$$f(t) = \int_{-\infty}^{\infty} \hat{f}(k)\, e^{ikt}\, dk,$$

$$\boxed{\hat{f}(k) = \frac{1}{2\pi}\int_{-\infty}^{\infty} f(t)\, e^{-ikt}\, dt}$$

A new problem:

Suppose we have a discrete set of N data points $y_0, y_1, \dots, y_{N-1}$ sampled over equal time increments Δt. Define

$$t_n = \underbrace{\Delta t}_{\text{Fixed}}\, n$$



Fit the data with a sum of sines & cosines (or really with complex exponentials using a complex Fourier series.) That is, determine all of the "frequencies" in this discrete set of data points.

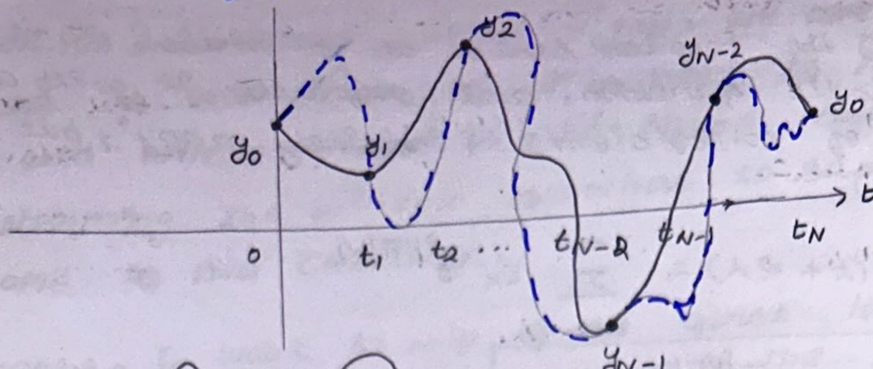
Step:1: Make it periodic.

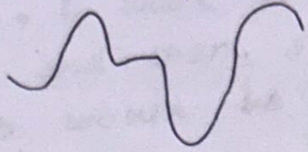
"Add another point t_N and define $y_N = y_0$

Example: 3.1

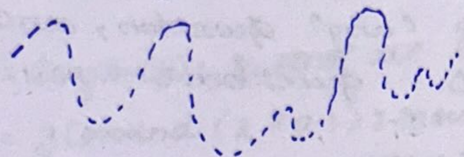
Let $f(t)$ and $g(t)$ be two functions that fit all of the data $y_0, \dots, y_{N-1}$ and satisfying

$f(t_N) = f(t_0)$, $g(t_N) = g(t_0)$





$\rightarrow f(t)$ function (going through all data points)



$\rightarrow g(t)$ goes through the same data points.

we can extend $f(t)$ and $g(t)$ to functions that are periodic of period $\Delta t\, N$.

This implies that we can write

> why t_n? in $e^{ik\pi t_n/N\Delta t}$
>
> $t_n \rightarrow$ sampled time

$$f(t) = \sum_{k=-\infty}^{\infty} c_k\, e^{ik\pi t/(t_N/2)} = \sum_{k=-\infty}^{\infty} c_k\, e^{\frac{i2\pi kt}{N\Delta t}}$$

> why $\frac{t_N}{2}$ = we will take N from the period e^{2N}

$\boxed{t_N = 2L}$

$\boxed{\frac{2N}{2} = N}$

> why $e^{ik\pi t} = \cos(k\pi t) + i\sin(k\pi t)$

for some set of complex co-efficients c_k.

$$g(t) = \sum_{k=-\infty}^{\infty} c_k'\, e^{i2\pi kt/N\Delta t}$$

for a set of co-efficients $c_k' \neq c_k$.

The problem: Any way of filling in the space b/w the points leads to a new function, with a new set of co-efficients. Thus there are infinitely many ways to fit the data with sines and cosines (or equivalently, with complex exponentials as we are actually doing here.

modified goal: represent the data as simply as possible with a continuous function obtained as a sum of complex exponential.

Observe that when we compare complex exponentials only at the discrete times where we sampled our data,

$$e^{i2\pi k t_n/N\Delta t} = e^{i2\pi k(n\Delta t)/N\Delta t} = e^{i2\pi kn/N}$$

$\therefore \boxed{t_n = n\Delta t}$

$\therefore$ we can't distinguish the difference b/w the kth and k+N th frequency

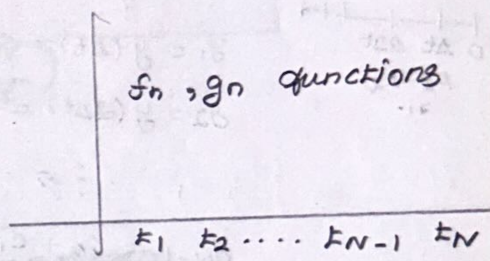
$$e^{i2\pi(k+N)t_n/N\Delta t} = e^{i2\pi kn/N}\cdot e^{i2\pi n} = e^{i2\pi kn/N}$$

$\boxed{\therefore t_n = n\Delta n}$

$$\boxed{\therefore e^{i2\pi n} = \cos 2\pi n + i\sin 2\pi n = 1}$$

t_1 to t_N → our system signal

t_n → sampled time. → $n\Delta t$

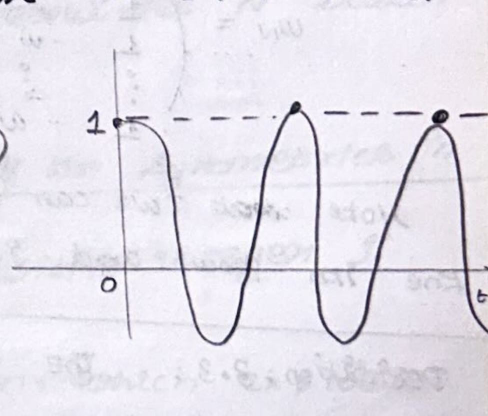


Literally, $e^{i2\pi kt/N\Delta t} = e^{i2\pi(k+N)t/N\Delta t}$

$$e^{i2\pi kn/N} = e^{i2\pi(k+N)n/N}$$

Consider the set of data that are all equal to 1.

Sampled data all equal to 1. Both the constant function 1 and $\cos(2\pi t/\Delta t)$ (Dashed & solid line) pass through all data.



Represented as the real parts of complex exponentials, the dashed and solid line is like is e^{i0}, and

$$e^{i2\pi t/\Delta t}.$$

- - - - → constant function 1

There are multiple frequencies that fit the data. Any frequencies higher than the sampling frequency can't be detected in the data.

Because the kth and the $(k+N)$th modes are equal when evaluated on the discrete time t_k, we can represent the data uniquely by the discrete time Fourier series.

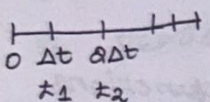
$$y(t) = \sum_{k=0}^{N-1} c_k \, e^{\frac{i2\pi k t}{N\Delta t}}$$

for some complex coefficients $c_0, c_1, \ldots, c_{N-1}$

so how do we actually compute these coefficients:

Define $\boxed{\omega = e^{i2\pi/N}}$, which is the Nth root of unity, where we have N data points.

Let's look at how our data points are represented in terms of ω.



$$y_0 = y(0) = c_0 + c_1 + c_2 + \ldots + c_{N-1}$$

$$y_1 = y(\Delta t) = c_0 + c_1 \omega^1 + c_2 \omega^2 + \ldots + c_{N-1} \omega^{N-1}$$

$$y_2 = y(2\Delta t) = c_0 + c_1 \omega^2 + c_2 \omega^4 + \ldots + c_{N-1} \omega^{2(N-1)}$$

$$\vdots$$

$$y_{N-1} = c_0 + c_1 \omega^{N-1} + c_2 \omega^{2(N-1)} + \ldots + c_{N-1} \omega^{(N-1)(N-1)}$$

Therefore we can write this data as a matrix equ.

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix} = W_N \begin{pmatrix} c_0 \\ \vdots \\ c_{N-1} \end{pmatrix}$$

$$W_N = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega & \omega^2 & \cdots & \omega^{N-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{N-1} & \omega^{2(N-1)} & \cdots & \omega^{(N-1)(N-1)} \end{pmatrix}$$

Note that we can describe this as the co.efficients in the ith row and jth column is $(W_N)_{ij} = \omega^{(i-1)(j-1)}$

Definition: 3.3: The inverse of the matrix is the N point Fourier matrix.

$$W_N^{-1} = F_N = \frac{1}{N} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \bar{\omega} & \bar{\omega}^2 & \cdots & \bar{\omega}^{N-1} \\ 1 & \bar{\omega}^2 & \bar{\omega}^4 & \cdots & \bar{\omega}^{2(N-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \bar{\omega}^{N-1} & \bar{\omega}^{2(N-1)} & \cdots & \bar{\omega}^{(N-1)(N-1)} \end{pmatrix}$$

Def: 3.4: we say that the vector

$$y = WC$$

$$\boxed{W^{-1}(y) = c}$$

$$\begin{pmatrix} c_0 \\ \vdots \\ c_{N-1} \end{pmatrix} = F_N \begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

is the Finite discrete Fourier transform (FDFT) of

$$\begin{pmatrix} y_0 \\ \vdots \\ y_{N-1} \end{pmatrix}$$

(Note: The discrete Fourier transform is the periodic, continuous functions obtained by imaging that we take these co-efficients and the sum of continuous functions, allowing it to be defined for all times t.)

$$\omega = e^{i2\pi/N}$$

$$\boxed{\begin{aligned} \therefore\ y_0 &= y(0) = c_0 + c_1 + \cdots + c_{N-1} \quad (n=0) \\ \therefore\ y_1 &= y(\Delta t) = c_0 + c_1\omega + \cdots + c_{N-1}\,\omega^{N-1} \quad (n=1) \end{aligned}}$$

$$y(t) = \sum_{k=0}^{N-1} c_k\, e^{\left(\frac{i2\pi kt}{N\Delta t}\right)}$$

Fast fourier transform

The Fast Fourier Transform (FFT) is an algorithm originated by cooley & Tuskey in 1965 for Computing the FDFT very efficiently.

Note that if A is an $N \times N$ matrix, and y is an $N \times 1$, to compute Ay requires N^2 scalar multiplications in general.

we reduce this taking advantage of the symmetries in F_N:

suppose that $N = 2^i$ for a positive integer i.

FFT is a divide & conquer algorithm, which exploits the fact that

$$F_N y = \begin{pmatrix} I_{N/2} & D_{N/2} \\ I_{N/2} & -D_{N/2} \end{pmatrix} \begin{pmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{pmatrix} \begin{bmatrix} y_1 \\ y_3 \\ \vdots \\ y_{\frac{N}{2}-1} \\ y_2 \\ \vdots \\ y_{N/2} \end{bmatrix}$$

Splitting $N \times N$ matrix (F_N) in to two matrices.

$I_{N/2}$ is the $\frac{N}{2} \times \frac{N}{2}$ identity matrix, and $D_{N/2}$ is the diagonal matrix.

$$D_{N/2} = \begin{bmatrix} 1 & & & & \\ & \bar{w} & & & \\ & & \bar{w}^2 & & \\ & & & \ddots & \\ & & & & \bar{w}^{N/2} \end{bmatrix}$$

we continue to apply the same rule to the $F_{N/2}$ matrices to get a formula for F_N involving four $F_{N/4}$ matrix multiplications instead.

we can compute $F_N y$ with only $\frac{1}{2} N g = \frac{1}{2} N \log N$ multiplications!

$$\hat{f} = F_N y = \begin{bmatrix} I_{N/2} & -D_{N/2} \\ I_{N/2} & -D_{N/2} \end{bmatrix} \begin{bmatrix} F_{N/2} & 0 \\ 0 & F_{N/2} \end{bmatrix} \begin{bmatrix} f_{even} \\ f_{odd} \end{bmatrix}$$

say

$N = 2^g$

$N = 2^{10}$

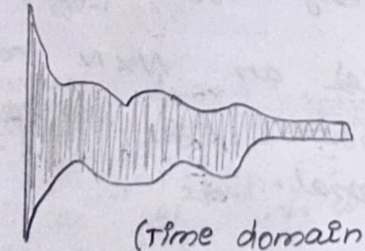
$\boxed{N = 1024}$

$$\boxed{F_{1024} \rightarrow F_{512} \rightarrow F_{256} \rightarrow F_{128} \rightarrow F_{64} \rightarrow \cdots \rightarrow F_4 \rightarrow F_2}$$

DFT in MATLAB

Tune

when the string is stretched



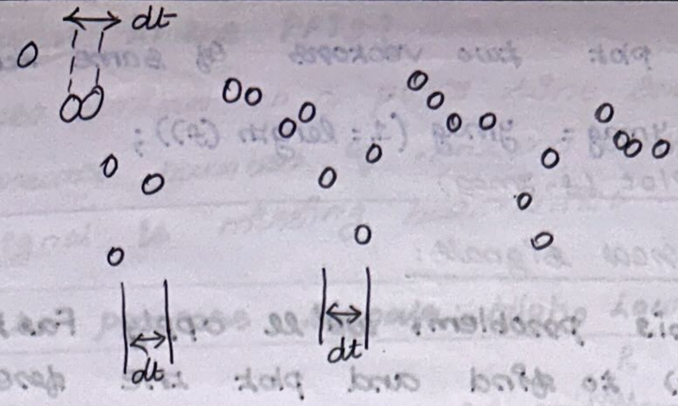
(Time domain)

'How we can tell that the string is in tune?'

Some have earphones. We need to analyze.

• How we can obtain & plot the frequency info for real valued function?

Though sound signals are continuous, for storing them digitally, need to sample them discretely at regular intervals.



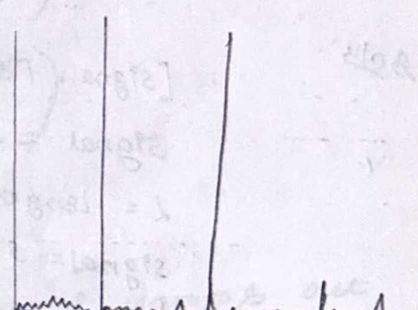
$$\Delta t = \frac{1}{FS}$$

FS → sampling frequency.

Matlab

y = fft (signal); → output is a complex vector as of our original signal.

ymag = abs (y) → To create vector magnitudes in frequency domain.



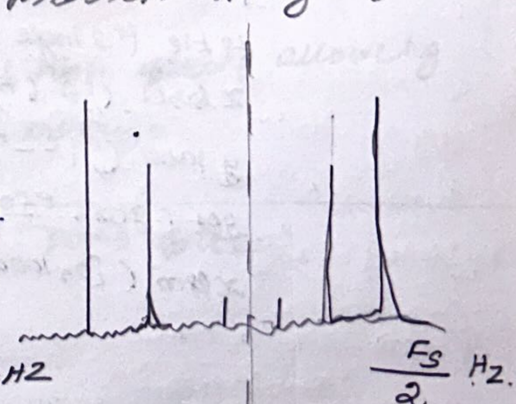
we found the Amplitude (magnitude)

Now: need to match them with corresponding frequencies.

N = length (ymag);

The separation b/w points in the frequency domain is $\frac{FS}{N}$. The highest frequency that can be extracted from the time domain data is $\frac{FS}{2}$ (Nyquist) frequency.

For real valued data like our sound signal, the F.S is the mirror image of the right side of the left.



Create : vector represents the frequencies using colon operator.

Start at zero

↑ by FS/N

End point $\frac{FS}{2}$.

we can only plot two vectors of same length.

```
>> ymag = ymag (1: length (f));
plot (f, ymag)
```

Plotting FFT for real signals:

In this problem, we'll apply Fast Fourier Transform (FFT) to find and plot the frequencies of our guitar sound signal. The template code extracts the signal from 30% to 50% of the original data. Follow the procedure in the video, to plot the single sided frequency spectrum.

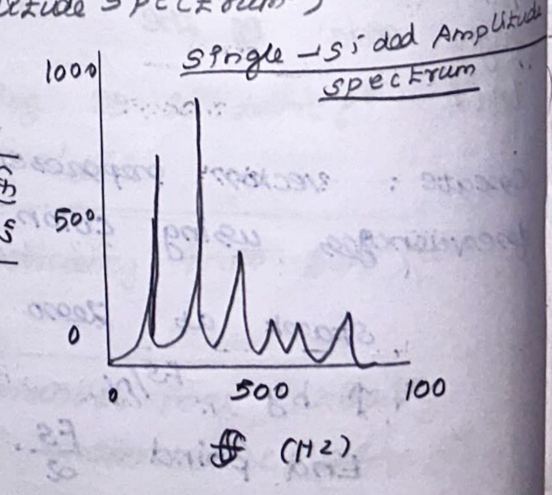
1) Find F.T of signal & store in y.

2) calc the single-sided magnitude spectrum and store in yMag.

3) create a vector, f, representing the frequencies of ymag.

solu:

```
[signal, Fs] = audioread ('1803_music data - guitar1.wav');
signal = signal (:, 1); → Taking only one channel.
L = length (signal);
signal = signal (round (L* 0.3): round (L* 0.5));
y = fft (signal);
ymag = abs (y);
N = length (ymag)
f = 0: Fs/N: Fs/2.
ymag = ymag (1: length (f));
plot (f, ymag, '-*');
title ('single - sided Amplitude spectrum')
xlabel ('f (Hz)')
ylabel ('|c_n (f)|')
set (gca, 'fontsize', 18)
xlim ([0, 1000])
```



What's different in the FFTS?

Most similar to a pure sine sound wave – Guitar

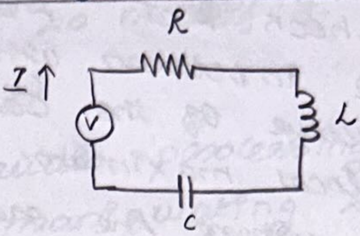
Greater number of nonzero harmonics – Voice

Signal is missing harmonics – Clarinet.

How to process signals: High, Low, & mid pass Filters

How to process?

RLC



R – Resistance in ohms

L – Inductance in Henry

C – capacitance in Farads

V – voltage source (volts)

If $V(t) = \cos(\omega t)$

$$V_R(t) = \mathrm{Re}\left(\frac{iR\omega}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega}\, e^{i\omega t}\right)$$

$$V_L(t) = \mathrm{Re}\left(\frac{-L\omega^2}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega}\, e^{i\omega t}\right)$$

$$V_C(t) = \mathrm{Re}\left(\frac{\frac{1}{C}}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega}\, e^{i\omega t}\right)$$

Low pass filter:

System that takes in a signal, & damps out high frequency signals, allowing the low frequency signals to pass through.

High pass filter:

System that takes in a signal, & damps out low freq signals, allowing high freq signals to pass

mid pass filter:

Damps out both low & high freq, allowing mid range freq signals to pass through

V_R – system response. → mid pass filter

V_L → High pass filter

V_C → Low pass filter.

$$\left| G_R(\omega) e^{i\omega t} \right| = \left| \frac{iR\omega}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega} \right|$$. Observe that for ω near zero, the complex gain has magnitude zero as well.

For ω large, the complex gain is zero as well. We can check that there is a finite value ω_r somewhere in between where this +ve function (the absolute value of the complex gain) must be +ve and have a local maximum. Thus if we consider a series RLC ckt where the response is the voltage drop across the resistor, this has the effect of damping out low & high frequency inputs, and letting midrange signals through. Thus it's a mid pass filter.

$$\left| G_L(\omega) e^{i\omega t} \right| = \left| \frac{-L\omega^2}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega} \right|$$. Observe that for ω near zero, the magnitude of the response is also near zero, but as ω approaches ∞, the magnitude of the response is near 1. Therefore if we consider a series RLC ckt whose response is the voltage drop across the inductor, this has the effect of damping out low frequencies & letting high freq pass through. It's a high pass filter.

$$\left| G_C(\omega) e^{i\omega t} \right| = \left| \frac{\frac{1}{C}}{\left(\frac{1}{C} - L\omega^2\right) + iR\omega} \right|$$. Observe that $\omega \to 0$, the magnitude response is 1, but $\omega \to \infty$, the magnitude response is zero. Therefore if we consider a series RLC ckt whose response is Vdrop across the capacitor, this has the same effect of damping out higher freq & letting low frequency signals pass through. It's a low pass filter.

Resistor	C	L
$\omega \to 0$ $R = 0$ } mid.	$\omega \to 0$ $R = 0$	$\omega \to 0$ $R = 1 \to$ low
$\omega \to \infty$ $R = 0$	$\omega \to \infty$ $R = 1$ $\hookrightarrow$ high	$\omega \to \infty$ $R = 0$

Manipulating spectrum data

we will see how we can manipulate the data in the frequency spectrum and go back to a sound signal using the inverse FFT.

y = fft (signal);

% Do something to modify y.

new signal = ifft (y)

This is essentially how all audio processing works. Audio processing works by manipulating signals in the frequency spectrum. The fact that we can get back to the time series - and audio signal - is what makes this work!

outline of how this process work?

1) Listen to your signal. Oh. no! something is wrong.

2) Take the FFT of your signal, and look at the single sided frequency spectrum. See what it can tell you about what's wrong with the signal.

3) Now recreate the symmetric signal (undoing this processing needed to view single sided spectrum) coming from a real sound wave

4) Take the inverse FFT to get a signal back. Try listening to see if it sounds better.

```
[y,FS] = audioread ('1803 - musicdata - voice 1.wav');
y = y(:,1);  (% Take first channel of signal)
sound (y, FS).
n = length (y);   % Length of signal.
t = (0:n-1) * (1/FS);  (% Time series vector)

figure (1)
plot (t, y)
```

```
Y = fftshift (fft (y));          % Take Fourier series & take a symmetric shift
fshift = (-n/2 : n/2-1) * (FS/n);
            % determine the frequency vector
L = length (fshift);     % Find the length of frequency values
yfilt = abs(Y);
```

---

method : 1

```
plot (fshift, abs(yfilt))
minYfilt = find (fshift < -0.3);
maxYfilt = find (fshift < 0.3);
index1 = max (minYfilt)
index2 = max (maxYfilt)
```

(or)

method : 2

```
middle = L/2;

hertz300 = middle/100;
index1 = round (middle - hertz300);
index2 = round (middle + hertz300);
```

Fading middle region of the signal

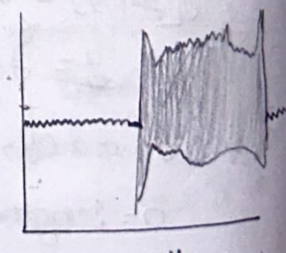
```
yfilt (index1 : index2) = 0
```

% Take the inverse Fourier (FFT) to create a filtered sound signal.

```
soundfilt = ifft (ifftshift (yfilt), 'symmetric');
sound (soundfilt, FS)
```

---

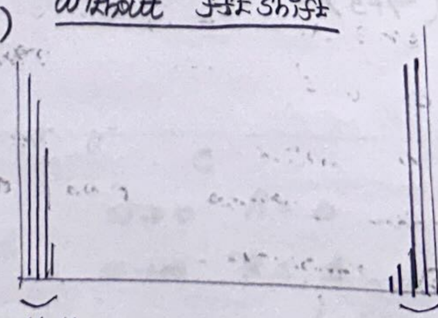
output:

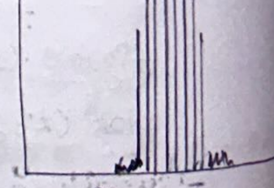


fftshift → shift zero frequency component to the middle of the spectrum.

$fshift = (-\frac{n}{2} : \frac{n}{2} - 1) * (FS/n)$

↳ For x axis intervals (vs) our final signal





More signal processing

We have looked at extremely simple ideas coming from wired telephones & AM radios, as well as thinking about using an RLC circuit to filter signals. We can imagine that the world we live in is far more complicated. The circuits involved in our cell phone to filterout all kinds of different noices & signals - wireless internet, cellular data & GPS for example - are much more complicated than a simple RLC circuit. But nonetheless, the underlying principles are the same.

All of the exercises you've done so far involving the discrete Fourier transform to study sound signals coming from musical instruments & voices playing a single note. These signals didn't change much in time. However, we interact daily with speak recognition software that is able to process sound signals in real time fairly accurately (using machine learning algorithms among other things.) spectrograms are a visual way of interpreting the Discrete Fourier transform of signals that change in time.

Spectrograms

url = 'https://courses.edx.org/asset-v1:MITx+18.03Fx+3T2018+type@asset+block@spectrogramExample.mlx';
websave ('spectrogramExample.mlx', url) → Spectrogram.mlx.

Spectrogram:

Divide up the signal into small chunks & perform a Fourier transform on each chunk. Well, good - we can create an image where

one axis → Frequency info.

Second axis → Time

color - power (Intensity of the freq) of the frequency at that time. This image is called a spectrogram.

Solving ODEs with Fourier series

1) Apply F.S. methods to solve LTI diff eqn with general periodic input signals.

2) Describe the system response to a general periodic input signal in terms of a Fourier Series.

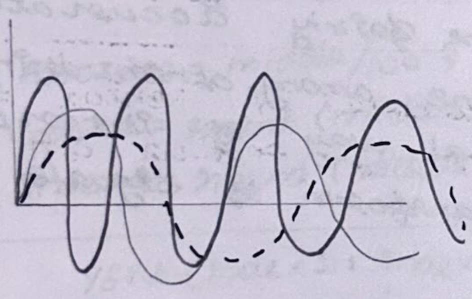
3) Use resonance to determine the dominant Fourier co.effs in a system response to an input signal described as a Fourier series.

4) Think of sound as a superposition of sine waves, & understand the ears ability to pick out Fourier co.effs, of the waves as rather than hearing the superposition as one object.

---

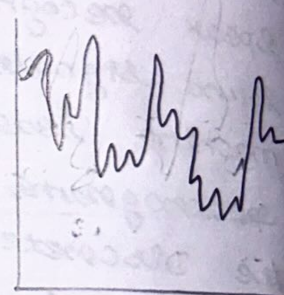
pure musical tone have pure osciuation in terms of vibrations.

suppose triads (3 signals) with their own osciuation periods.



C + E + G

we hear → their sum



(mess)

Tone deaf → Insensitive to some pitches. (Differences in pitch).

$f(t) = \sin\omega_1 t + \sin\omega_2 t + \sin\omega_3 t.$

Is our brain has an Integrator? calculating the co.efficients ? NO! Then how we are hearing triads (as their sum)

ERF

p - polynomial with real, constant coeff, $D = \frac{d}{dt}$ a diff operator, and r a (real (or) complex number). If $P(r) \neq 0$, then particular solution to the inhomogene data can

$P(D)y = e^{rt}$ is gn by $\boxed{y_p = \frac{e^{rt}}{P(r)}}$

Caveat (எச்சரிக்கை)

If $P(r) = P'(r) = P''(r) = \cdots = P^{(k-1)}(r) = 0$, but $P^{(k)} \neq 0$, then P.I to $P(D)y = e^{rt}$ is gn by

$$y_p = \frac{t^k e^{rt}}{P^{(k)}(r)}$$

Sinusoidal input:

$P(D)x = \cos(\omega t) \rightarrow$ Real part of $P(D)z = e^{i\omega t}$

$P(D)x = \sin(\omega t) \rightarrow$ Img part of $P(D)z = e^{i\omega t}$

$\therefore$ P.I to $P(D)x = \cos\omega t$ is gn by $x_p = Re\left[\frac{e^{i\omega t}}{P(i\omega)}\right]$

$\therefore$ P.I to $P(D)x = \sin(\omega t)$ is gn by $x_p = Im\left[\frac{e^{i\omega t}}{P(i\omega)}\right]$

Applications of Fourier series

$f(t)$ is an odd periodic function of period 2π. Find the periodic function $x(t)$ of period 2π that's a solution to

$$\ddot{x} + 50x = f(t)$$

Solu: $x(t) \rightarrow$ output, $f(t)$ Input.

Special case:

what's the system response to the input signal $\sin(nt)$? In other words, what's a solution to

$$\ddot{x} + 50x = \sin(nt)$$

with same (smallest) period as $\sin(nt)$?

Solu:

$$\ddot{z} + 50z = e^{int}$$

$$z = \frac{e^{int}}{(+in)^2 + 50} = \frac{1}{50 - n^2} e^{int}$$

Complex gain $= \frac{1}{50-n^2}$. Then,

$$x = \operatorname{Im}\left(\frac{1}{50-n^2} e^{int}\right)$$

$$= \frac{1}{50-n^2} \sin(nt)$$ is the system

response to $\sin(nt)$.

Input signal	System Response
e^{int}	$\frac{1}{50-n^2} e^{int}$
$\sin nt$	$\frac{1}{50-n^2} \sin nt$
$\sin t$	$\frac{1}{49} \sin t$
$\sin 2t$	$\frac{1}{41} \sin 3t$
$\vdots$	$\vdots$
$\sum_{n\geq 1} b_n \sin nt$	$\sum_{n\geq 1} \frac{1}{50-n^2} b_n \sin nt$.

System response

Suppose f is an odd periodic signal of period 2π. Since f is odd, the Fourier series of f is a linear combination of sine functions.

$$f(t) = b_1 \sin t + b_2 \sin 2t + b_3 \sin 3t + \dots$$

Let $f(t)$ be the input of the system,

$$\dot{x} + 50x = f(t)$$

From superposition principle, the system response of $f(t)$ is

$$x(t) = b_1\left(\frac{1}{49}\right) \sin t + b_2\left(\frac{1}{46}\right) \sin 2t + b_3\left(\frac{1}{41}\right) \sin 3t + \dots$$

Note: Each Fourier component $\sin nt$ has a different gain: the gain depends on the frequency.

we can, write P.I

$$\boxed{x_p(t) = \sum_{n\geq 1} \frac{1}{50-n^2} b_n \sin nt}$$

Near resonance

Problem: 5.1:

If the system response of $\sin nt$ is $\dfrac{\sin(nt)}{50-n^2}$, for which input signal $\sin(nt)$ is the gain the largest?

Solu: The gain is $\left|\dfrac{1}{50-n^2}\right|$, which is largest when the $|50-n^2|$ is smallest.

$$50-n^2=0$$
$$n^2=50$$
$$n=7.071067812$$
$$\boxed{n\approx 7}$$

The gain for $\sin 7t$ is 1, and the next largest gain occurs for $\sin 6t$ and $\sin 8t$ is $\frac{1}{14}$.

$$\boxed{\therefore \left|\frac{1}{14}\right| \text{ and } \left|-\frac{1}{14}\right|}$$

. Thus the system approximately filters out all the Fourier components of $f(t)$ except $\sin 7t$ term.

5.2: Let $x(t)$ be the periodic solution to

$$\ddot{x}+50x=\frac{\pi}{4}\,\mathrm{sq}(t)$$

which Fourier coefficient of $x(t)$ is largest? which is second largest.

Solu:

$$\therefore \mathrm{sq}(t)=\frac{4}{\pi}\sum_{n\text{ odd}}\frac{\sin nt}{n}$$

$\therefore n \Rightarrow$ Integer

$$\frac{\pi}{4}\,\mathrm{sq}(t)=\sum_{n\geq 1,\text{ odd}}\left(\frac{1}{50-n^2}\right)\frac{\sin nt}{n}$$

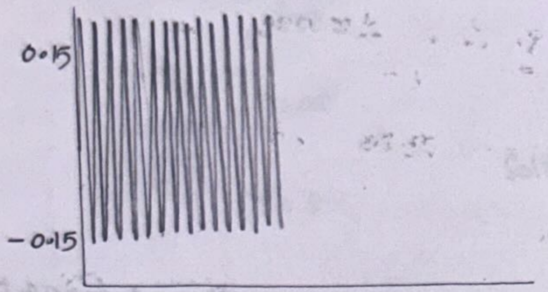
$$=0.020\sin t+0.008\sin 3t+0.008\sin 5t+$$
$$0.143\sin 7t-0.004\sin 9t-$$

(even smaller)

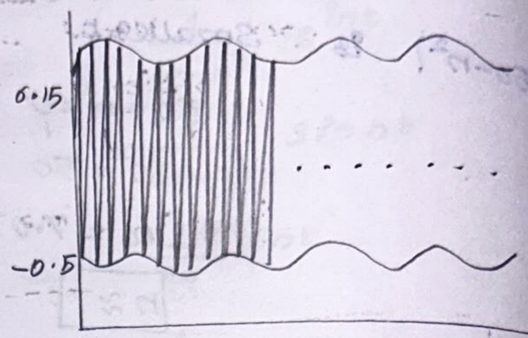
So the co-efficient of $\sin 7t$ is largest & the co-eff of $\sin t$ is second largest.

$\therefore$ (Fourier co.eff is large when $\left(\frac{1}{(50-n^2)n}\right)$ is large only when one of n or $50-n^2$ is small.

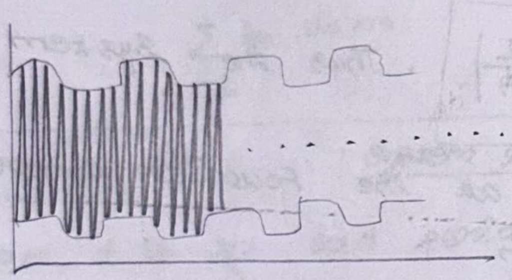
Remark: 5.3: Even though, the system response is a complicated Fourier series, with infinitely many terms, only one or two are significant, and the rest are negligible.



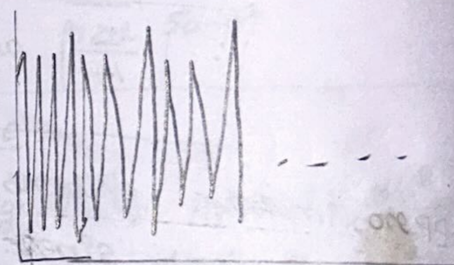
sum of largest term



sum of 1st 2 largest terms



Sum of 1st five terms



Sum of 1st largest terms

Components of $n \neq 7$ (sinnt) are greatly reduced

Pure resonance

what happens if instead of considering the diff eqn

$$\ddot{x} + 50x = \frac{\pi}{4} sq(t),$$

we change 50 to 49

$$\ddot{x} + 49x = \frac{\pi}{4} sq(t)$$

∞ many solutions, but none of them are periodic. $n \neq 7$, we can solve $\ddot{x} + 49x = \sin nt$ using ERF. since in is not a root of $r^2 + 49$

For $n=7$, we can still solve $\ddot{x}+49x=\sin 7t$ (the existence & uniqueness theorem guarantees this), but the solution requires generalized ERF, and involves t, and hence is not periodic: it turns out that one solution is $-\frac{t}{14}\cos 7t$.

$$y_p = \frac{t\, e^{7t(i)}}{p'^{(1)}(r)} = \frac{t e^{7it}}{-2r} = -\frac{t}{14}\cos 7t$$

(Real part)

For the input signal $Sq(t)$, we can find a solution x_p by superposition: most of the terms will be periodic, but one of them will be $\frac{1}{7}\left(-\frac{t}{14}\cos 7t\right)$, and this makes the whole solution x_p non-periodic.

There are ∞ many solution, which differ by adding the homogeneous solution, namely

$$x_p + c_1\cos 7t + c_2\sin 7t \quad \text{for any } c_1 \text{ and } c_2.$$

These solutions still includes the $\frac{1}{7}\left(-\frac{t}{14}\cos 7t\right)$ term hence not periodic.

If the ODE had been

$$\ddot{x} + 36x = \frac{\pi}{4} Sq(t)$$

then all solutions would have been periodic. $\frac{\pi}{4}Sq(t)$ has no $6t$ term in its Fourier series. In general, for a periodic function f, the ODE $P(D)x = f(t)$ has a periodic solution if & only if for each term $\cos\omega t$ or $\sin\omega t$ appearing with a non-zero co.eff in the Fourier series of f, the number $i\omega$ is not a root of $P(r)$.

Period 2L

$$x'' + \omega_0^2 x = f(t) \quad \text{(undamped case)}$$

Can find x_p if R.H.S $\begin{cases}\cos\omega t \\ \sin\omega t\end{cases}$ (or) some multiples of them.

The Gain will be $\left(\frac{1}{\omega_0^2-\omega^2}\right)$

$\omega_0 \rightarrow$ natural frequency

$\omega \rightarrow$ Imposed frequency

Let

$f(t)=\left(\frac{1}{2}+\frac{2}{\pi}\sum_{odd}\frac{\sin n\pi t}{n}\right)$ [say then generalize]

$f(t)=\frac{a_0}{2}+\sum(a_n\cos\omega_n t+b_n\sin\omega_n t)$

To make the general period to be $2L$:

Comparing,

$$\boxed{\omega_n=\frac{n\pi}{L}}$$

$\boxed{\therefore \text{ period of } f(t) \text{ is } 2L}$

By linearity.

$$x_p=\left(\sum_1^\infty\frac{a_n\cos\omega_n t}{\omega_0^2-\omega_n^2}+\frac{b_n\sin\omega_n t}{\omega_0^2-\omega_n^2}\right)+\frac{a_0}{2\omega_0^2}$$

$\boxed{\therefore (\omega_n=0) \rightarrow \text{at } a_0 \text{ case}}$

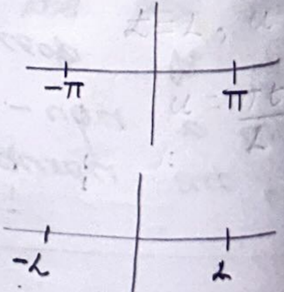
$\therefore f(t)$ has a period of $2L$

$$f(t)=\frac{1}{2}+\frac{2}{\pi}\sum_{odd}\frac{\sin n\pi t}{n}$$

$$f(t)=\frac{a_0}{2}+\sum(a_n\cos\omega_n t+b_n\sin\omega_n t)$$

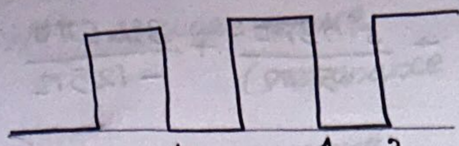
$\therefore \boxed{\omega_n=\frac{n\pi}{L}} \rightarrow$ Normally

(see previous problems)



Resonant response of a step wave

$$f(t)=\begin{cases}1 & 0<t<1\\ 0 & -1<t<0\end{cases}$$ function of period 2



$$f(t) = \frac{1}{2} + \frac{2}{\pi} \sum_{n\,odd} \frac{\sin(n\pi t)}{n}$$

Q.1: $\ddot{x} + 100x = f(t)$

solu:

$$x_p(t) = \frac{a_0}{2\omega_0^2} + \sum_{n \geq 1} \frac{a_n \cos \omega_n t}{\omega_0^2 - \omega_n^2} + b_n \frac{\sin \omega_n t}{\omega_0^2 - \omega_n^2}$$

$$\therefore \quad x_p = \frac{1}{2 \times 100} + \left(\sum_{n\,odd} \frac{\sin(n\pi t)}{(100 - (n\pi)^2) n} \right) \frac{2}{\pi}$$

$$= \frac{1}{200} + \frac{2}{\pi} \sum_{n\,odd} \left(\frac{\sin(n\pi t)}{(100 - (n\pi)^2) n} \right)$$

Largest amplitude:

$$100 - (n\pi)^2 = 0$$

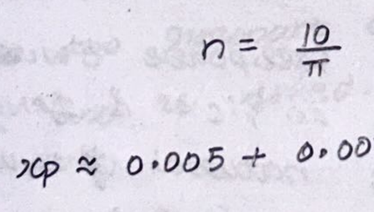
$$100 = n^2 \pi^2$$

$$n^2 = \frac{100}{\pi^2}$$

$$n = \frac{10}{\pi} = 3.1830 \approx 3$$

$$x_p \approx 0.005 + 0.007 \sin(\pi t) + \underbrace{0.019 \sin(3\pi t)}_{Large} - 0.0009 \sin(5\pi t)$$

Response



(Input $f(t)$) $\longrightarrow$ $x'' + \omega_0^2 x = f(t)$

Here, $\boxed{\omega_n = n\pi}$

Response:

$$x_p(t) = \frac{1}{2\omega_0^2} + \sum_{n\,odd} \frac{\sin n\pi t}{(\omega_0^2 - (n\pi)^2)} \cdot \frac{1}{n}$$

(size of the coefficients: our interest)

Let natural frequency $\omega_0 = 10$. (approximation)

$$x_p(t) \approx 0.005 + 0.6\left(\frac{\sin \pi t}{91\,n} + \frac{\sin 3\pi t}{15\,n} + \frac{\sin 5\pi t}{-125\,n} + \cdots\right)$$

(Assuming)

What's clear?

The coefficients are going toward -ve values. (decreasing).

$$\approx 0.005 + \frac{0.1}{n}\sin \pi t + \frac{0.01}{n}\sin \pi t - \frac{0.05}{n}\sin 3\pi t - \frac{0.01}{n}\sin\left(\frac{5\pi}{2}\right) + \cdots$$

The frequencies which make up the response don't occur with the same amplitude.

$\boxed{0.05 \text{ is very larger than any of them}}$

So 0.05 is the large, $\sin 3\pi$ is the main one.

"very near resonance"

We can decompose a function (periodic: most efficiently) in to sum of frequencies of sine & cosines

Fourier analysis

But the system is not going to response equally for all frequencies. It's going to pick & favour the one which is closest to its natural frequency.

- These frequencies & their importance are hidden until we do Fourier analysis & look out the size of Co.eff.

But the system does & encourage them.

Harmonic frequency mathlet

$$\ddot{x} + \omega_0^2 x = \omega_0^2 f(t) \longrightarrow \text{mathlet.}$$

- Harmonic frequency response:

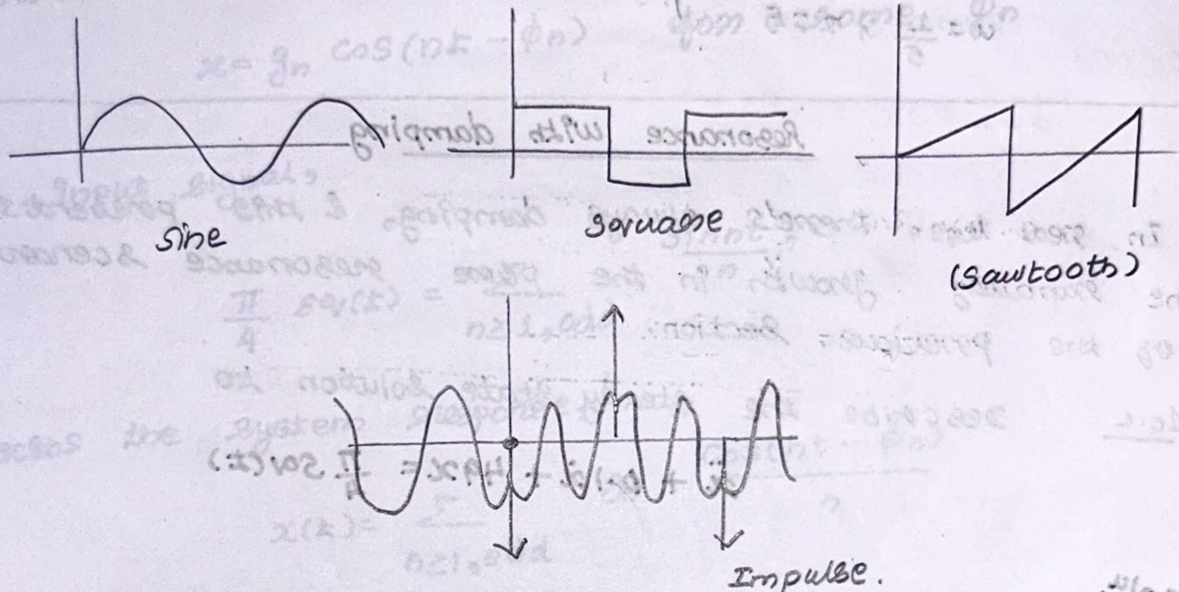
: variable natural frequency

$f(t) =$ sine wave (Resonance $w_n = 1$)

$f(t) =$ square wave
(resonance at $w_n = 1, 3, 5, \ldots$)

$f(t) =$ Sawtooth wave
(resonance at $w_n = 1, 2, 3, 4, 5, \ldots$)

$f(t) =$ Impulse wave
(resonance at $w_n = 1, 3, 5, \ldots$)



Resonant frequencies (variable Input frequency?

$$\ddot{x} + x = f(\omega t)$$

Here the resonant or natural frequency of this system is 1 and is fixed.

$f(wt) = \sin(wt)$

which of the frequencies (Angular) of the input are resonant with the system?

$w_n = 1.$

$f(wt) = Sq(wt)$

$$w = 1, \frac{1}{3}, \frac{1}{5}, \ldots$$

$$P.I = \ddot{x} + x = \sin(wt)$$

is gn by

$$x_p = \frac{\sin wt}{1 - w^2} \quad w \neq 1. \text{ (Resonance)}$$

P.I to

$$\ddot{x} + x = Sq(wt) = \frac{4}{\pi} \sum_{n \text{ odd}} \frac{\sin(wnt)}{n}$$

is

$$x_p = \frac{4}{\pi} \sum_{n \text{ odd}} \frac{\sin(wnt)}{(1-w^2n^2)n}$$

when $w=1$, the term $n=1$ is in resonance

$w = \frac{1}{3}$, $n=3$

$w = \frac{1}{5}$, $n=5$ } simply, $w = \frac{1}{n}$, n (Integer +ve)

Resonance with damping

In real life, there's always damping, & this prevents the runaway growth in the pure resonance scenario of the previous section.

10.2 Describe the steady state solution to

$$\ddot{x} + 0.1\dot{x} + 49x = \frac{\pi}{4} Sq(t)$$

Solu.

$0.1x$ is the damping term.

Remember: The steady state solution is the periodic solution (other solutions will be a sum of the steady state solution with a transient solution solving the homogeneous ODE

$$\ddot{x} + 0.1\dot{x} + 49x = 0$$

These transients solutions tend to 0 as $t \to \infty$. because the Co.effs of the char polynomial are +ve (In fact

Let's solve:

$$\ddot{x} + 0.1\dot{x} + 49x = \sin nt$$

$$\ddot{z} + 0.1\dot{z} + 49z = e^{int}$$

$$z_p = \frac{e^{int}}{(-n^2 + 0.1in + 49)}$$

$$z = \frac{1}{(49-n^2) + (0.1)ni} e^{(int)}$$