Handwritten Notes

Below are some handwritten notes that helped me in completing the project. They're ordered as follows.

#1: Runge-Kutta Method: As there were no chapters in Chopra that presented how to lay out the iteration loops for the Runge-Kutta Fourth-Order Method, a YouTube lecture (cited in the Report) showing a pseudocode was used extensively to create a script specific to single degree of freedom structures to dynamic response.

#2: Stiffness Matrix Condensing: The chimney displacement example presented in Chopra assumes that the reader knows how to derive a statically condensed stiffness matrix, and I go through the process in the notes of deriving the text's stiffness matrix, while noting the patterns in the matrix so that it is programmable depending on the amount of degrees of freedom the user chooses.

#3: Eigenvector calculations: Assuming that phi_11 equals 1, I found out by writing out the matrices by hand, that the rest of the phi's can be found by crossing out the 1^{st} row, and the 1^{st} column, and then inverting the matrix.

#4: Fast Fourier Transform: Concepts discussed with Prof. Loh on Fast Fourier Tansform, and Power Spectral Density Function have been neatly organized into a single sheet.

Runge-Kutta Method

$$dv1 = h * F(t, x, v)$$

me $\rightarrow$ $$dv(i) = \underbrace{dt * ((p_{i-1}/m) - rkc * v(i-1) - rkk * u(i-1))}_{F(t,x,v)}$$

$$\cancel{du_1(i) = \Delta t * v_{i-1}}$$

$$\cancel{dv_1(i) = \Delta t * \left(\frac{p_{i-1} - c * \dot{u}_{i-1} - k * u_{i-1}}{m}\right)}$$

$$\cancel{du_2(i) = \Delta t * (\dot{u}_{i-1} + \dot{u}}$$

$$du_{1i} = \Delta t * v_{i-1}$$

full step

$$dv_{1i} = \frac{\Delta t}{m} * \left[\frac{p_{i-1} - c * v_{i-1} - k * u_{i-1}}{\cancel{m}}\right]$$

$$du_{2i} = \Delta t * \left[v_{i-1} + \frac{dv_{1i}}{2}\right]$$

talk about linear interpolation

$$dv_{2i} = \frac{\Delta t}{m} * \left[\frac{1}{2\cancel{m}} * (p_i + p_{i-1}) - \frac{c}{\cancel{m}} * \left(v_{i-1} + \frac{dv_{1i}}{2}\right) - \frac{k}{\cancel{m}} * \left(u_{i-1} + \frac{du_{1i}}{2}\right)\right]$$

$$du_{3i} = \Delta t * \left(v_{i-1} + \frac{dv_{2i}}{2}\right)$$

$$dv_{3i} = \frac{\Delta t}{m} * \left[\frac{\left(\frac{1}{2} * (p_i + p_{i-1})\right)}{\cancel{m}} - \frac{c}{\cancel{m}}\left(v_{i-1} + \frac{dv_{2i}}{2}\right) - \frac{k}{\cancel{m}} * \left(u_{i-1} + \frac{du_{2i}}{2}\right)\right]$$

$$du_{4i} = \Delta t * (v_{i-1} + dv_{3i})$$

full step.

$$dv_{4i} = \frac{\Delta t}{m} * \left[\frac{p_i}{\cancel{m}} - \frac{c}{\cancel{m}} * (v_{i-1} + dv_{3i}) - \frac{k}{\cancel{m}}(u_{i-1} + du_{3i})\right]$$

$$du_i = [du_{1i} + du_{2i} + du_{3i} + du_{4i}]/6;$$

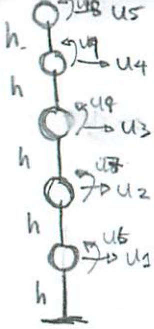
$$dv_i = [dv_{1i} + dv_{2i} + dv_{3i} + dv_{4i}]/6;$$

$$u_i = u_{i-1} + du_i$$

$\phi * g = u \rightarrow$ sn.

$\phi *$ ~~sk~~

Stiffness Matrix
Condensing



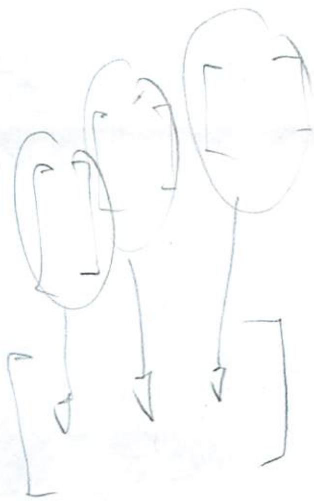
$$\begin{bmatrix} k_{tt} & k_{t0} \\ k_{0t} & k_{00} \end{bmatrix}$$

$k_{71} =$

$$\begin{array}{c|cccccccccc}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\
\hline
1 & \frac{24EI}{L^3} & \frac{-12EI}{L^3} & 0 & 0 & 0 & 0 & \frac{-6EI}{L^2} & 0 & 0 & 0 \\
2 & \frac{-12EI}{L^3} & \frac{24EI}{L^3} & \frac{-12EI}{L^3} & 0 & 0 & \frac{6EI}{L^2} & 0 & \frac{-6EI}{L^2} & 0 & 0 \\
3 & 0 & \frac{-12EI}{L^3} & \frac{24EI}{L^3} & \frac{-12EI}{L^3} & 0 & 0 & \frac{6EI}{L^2} & 0 & \frac{-6EI}{L^2} & 0 \\
4 & 0 & 0 & \frac{-12EI}{L^3} & \frac{24EI}{L^3} & \frac{-12EI}{L^3} & 0 & 0 & \frac{6EI}{L^2} & 0 & \frac{-6EI}{L^2} \\
5 & 0 & 0 & 0 & \frac{-12EI}{L^3} & \boxed{\frac{12EI}{L^3}} & 0 & 0 & 0 & \frac{6EI}{L^2} & \boxed{\frac{6EI}{L^2}} \\
6 & 0 & \frac{6EI}{L^2} & 0 & 0 & 0 & \frac{8EI}{L} & \frac{2EI}{L} & 0 & 0 & 0 \\
7 & \frac{-6EI}{L^2} & 0 & \frac{6EI}{L^2} & 0 & 0 & \frac{2EI}{L} & \frac{8EI}{L} & \frac{2EI}{L} & 0 & 0 \\
8 & 0 & \frac{-6EI}{L^2} & 0 & \frac{6EI}{L^2} & 0 & 0 & \frac{2EI}{L} & \frac{8EI}{L} & \frac{2EI}{L} & 0 \\
9 & 0 & 0 & \frac{-6EI}{L^2} & 0 & \frac{6EI}{L^2} & 0 & 0 & \frac{2EI}{L} & \frac{8EI}{L} & \frac{2EI}{L} \\
10 & 0 & 0 & 0 & \frac{-6EI}{L^2} & \boxed{\frac{6EI}{L^2}} & 0 & 0 & 0 & \frac{2EI}{L} & \boxed{\frac{4EI}{L}}
\end{array}$$

cancelled

$$\begin{array}{c|cccc}
 & \text{L } \Delta=1 & \text{L } \theta=1 & \text{R } \Delta=1 & \text{R } \theta=1 \\
\hline
\text{Shear @ L due to...} & \frac{12EI}{L^3} & \frac{6EI}{L^2} & \frac{-12EI}{L^3} & \frac{6EI}{L^2} \\
\text{Moment @ L due to...} & \frac{6EI}{L^2} & \frac{4EI}{L} & \frac{-6EI}{L^2} & \frac{2EI}{L} \\
\text{Shear @ R due to...} & \frac{-12EI}{L^3} & \frac{-6EI}{L^2} & \frac{12EI}{L^3} & \frac{-6EI}{L^2} \\
\text{Moment @ R due to...} & \frac{6EI}{L^2} & \frac{2EI}{L} & \frac{-6EI}{L^2} & \frac{4EI}{L}
\end{array}$$



phi(:,i) =

$$\begin{array}{cc}
\Delta=1 & \theta=1 \\
\frac{12EI}{L^3} & \frac{6EI}{L^2} \\
\frac{6EI}{L^2} & \frac{12EI}{L^3}
\end{array}$$

ktt
kt0
k00
k0t

$\hat{k} = k_{tt} - k_{0t} * k_{00}^{-1} k_{0t}$

$\frac{\cancel{12}EI}{L^3} - \frac{\cancel{6EI}}{\cancel{L^2}} * \frac{\cancel{L^3}}{\cancel{12EI}} * \frac{6EI}{\cancel{L^2}}$

$\frac{6EI}{L^3} - \frac{6EI}{L}$

Figuring out eigenvector (ϕ) vector calculations.

$\omega_1 = 1.7011$.

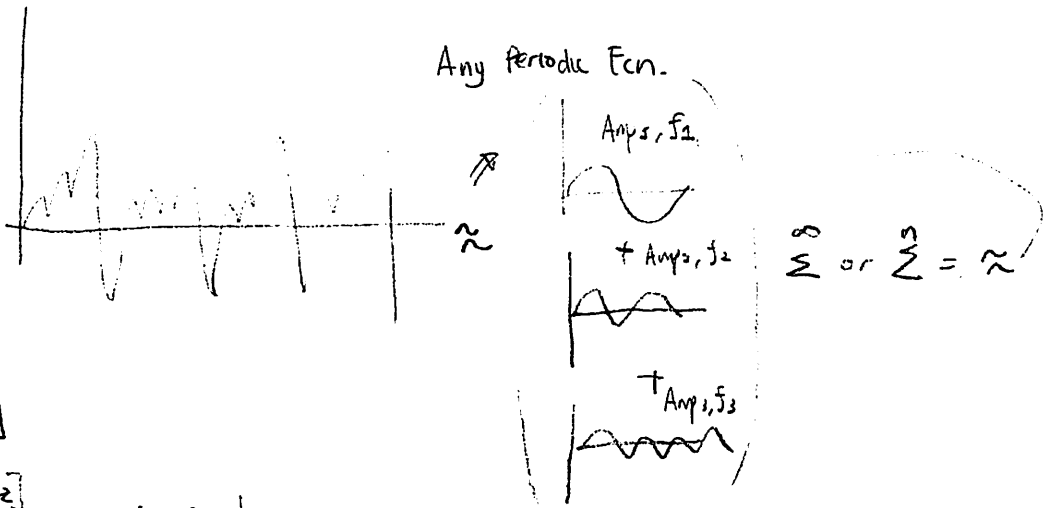
$$\left[K - \omega_1^2 M \right] \begin{Bmatrix} \phi_{11} \xrightarrow{} 1 \\ \phi_{21} \\ \phi_{31} \\ \phi_{41} \\ \phi_{51} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

$$\underbrace{\begin{bmatrix} 595322 & -376627 & 151062 & -37758 & 6295 \\ -376627 & 463028 & -338964 & 132199 & -22031 \\ 151062 & -338964 & 444355 & -301111 & 81845 \\ -37758 & 132199 & -301111 & 311998 & -115393 \\ 6295 & -22031 & 81845 & -115393 & 50575 \end{bmatrix}}_{C} \underbrace{\begin{Bmatrix} 1 \\ \phi_{21} \\ \phi_{31} \\ \phi_{41} \\ \phi_{51} \end{Bmatrix}}_{\phi} = \underbrace{\begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}}_{D}$$

FFT Fast Fourier Transform - Rewritten

Concept behind Fourier Series

Any Periodic Fcn.



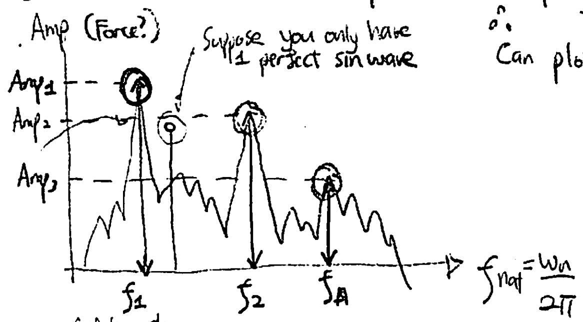
$\omega_n = \left[rad/s \right]$

$f_n = \frac{\omega_n}{2\pi} \left[1/s \text{ or } Hz \right]$

how many waves pass

Every sin curve has an amplitude & frequency

$\therefore$ Can plot how amplitude varies along frequency



bins

of time domain.

Matlab $\Rightarrow$ FFT(signal) $\Rightarrow$ produces complex #

$\# = a + bi$

WANT TO DO. look.

Power Spectral Density Function: allows you to plot real and complex value and just plot as frequency

It's discrete.: how to get Δf, range of freq, so on and so forth.

Should expect.:

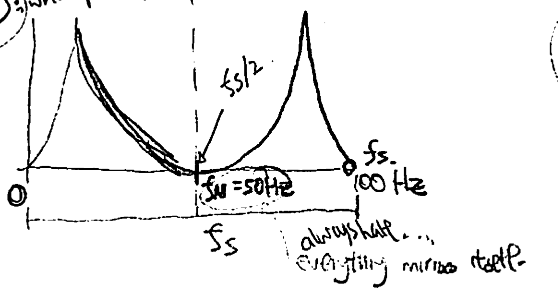
time step - discretizes. .

a lot has to do w. sampling rate $= \frac{1}{\Delta t} = 100\,Hz = f_s$

should be the norm of this..

Time step $= \Delta t$ $\quad 0.01s$

PSD: when plotted.



$f_N = \frac{1}{2} f_s$: Nyquist frequency

$\Rightarrow$ effectively only care about left half.