## Obtain first order approximation:

* Generate Y parameters out of these values:

Consider n=1:

At

Re-write this as:

To find the coefficients we can use the following:

Say B matrix contains the coefficients as follows:

So,

Code associated with this.

function [H\_impulse,num,deno]=first\_order\_approximation(realV,imagV,wo)

Yr = realV; % Real part of Y11

Yi = imagV; % Imaginary part of Y11

w = wo;

% Loop through each frequency point to construct A and C

A = [];

C = [];

% Loop through each frequency point to construct A and C

for k = 1:length(w)

wk = w(k);

Yr\_k = Yr(k);

Yi\_k = Yi(k);

% Construct rows for A and C

A\_row1 = [-1, Yr\_k]; % Real part

A\_row2 = [0, Yi\_k]; % Imaginary part

% Append to A

A = [A; A\_row1; A\_row2];

% C

C\_row1 = wk\* Yi\_k; % Real part

C\_row2 = -wk \* Yr\_k; % Imaginary part

% Append to C

C = [C; C\_row1; C\_row2];

end

% Solve for B = [a0; b0; a1; b1]

B = A \ C;

D=0;

% get cof

a0 = B(1);

b0 = B(2);

num = [a0];

deno = [1,b0];

% generated H

H\_impulse =@(s) (a0)./(s+b0);

end

clear

clc

% 2/s+2

f = 0:0.1:1;

w = 2\*pi\*f;

s = i\*w;

time = 5;

t = linspace(0, time, 1000);

exact = 2./(s+2);

exact\_impulse = 2\*exp(-2\*t);

[H,num,deno]=first\_order\_approximation(real(exact),imag(exact),w);

[A,B,C,D] = create\_state\_space(num,deno);

[y\_impulse, ~, ~, t] = AWE2(A, B, C, D, w(1), 1, time);

plot(t,exact\_impulse,t,y\_impulse,"--")

grid on

Tested and achieved error of 1.3043e-16:

## Plot data line given points:

Frequency response:

A graph of a function

AI-generated content may be incorrect.

Step response:

A graph with a line

AI-generated content may be incorrect.

Code:

clear;

clc;

step\_data = load('step');

vo = step\_data(:,2);

time = step\_data(:,1);

line\_data = load("data\_line");

f = line\_data(:,1);

v = line\_data(:, 2) + 1i\*line\_data(:, 3);

%{

plot(time,vo);

xlabel('time (s)')

grid on

title('step response')

%}

plot(f,abs(v));

xlabel("Frequency (Hz)")

ylabel("|Vo|")

grid on

title('Frequency response')

## Generate a model using iterative curve-fitting: