

```

% Lab 03

% H = e^(-yt)acos(wt)
% H = height above rest, y = damping coefficient
% a = starting height before release, w = frequency of oscillation
% t = time from release

% t vector : 0 - 4 seconds
% a = 1
% w = 6

clc
clear
close all
%% b.- e
dcoef = -1;
while dcoef < 0
    % ask user to enter starting damping coefficient
    dcoef = input('Please enter a starting damping coefficient: ');
    if dcoef < 0 % check if value is positive (while loop keeps going)
        fprintf('You entered %i Please enter a positive starting damping
coefficient.\n\n',dcoef);
    else % else print damping coefficient
        fprintf('You entered %i \n\n',dcoef); % \n = new line %i = dcoef
    end
end

numplots = -1;
while numplots < 0
    % ask user to enter number of damping coefficients to be plotted
    numplots = input('Please enter the number of damping values to plot: ');
    numplots = round(numplots); %rounding number of plots
    if numplots < 0
        % check if greater than zero
        fprintf('You entered %i Please enter a a number greater than 0.\n\n
n',numplots);
    else
        % else print
        fprintf('You entered %i \n\n', numplots);
    end
end

coeffs = linspace(dcoef,numplots*dcoef,numplots); % vector of evenly spaced dcoefs
t=linspace(0,4,100); % time vector | ranges from 0 to 4 | contains 100 evenly
spaced values

%% f. Calculate and plot the height over time for each of the damping coefficients
% a. Use a for-loop that runs from 1 to the length of the damping coefficient
vector
for y=1:length(coeffs)
    h = CalcPosition(t, coeffs(y)); % b. compute height for each damping
coefficient
    switch y % c. switch statement to assign a character array to color & linetype
code
        case 1
            LineColor='b-'; % 5 distinct options
        case 2
            LineColor='r--';
        case 3

```

```

        LineColor='xm';
    case 4
        LineColor='g-.';
    case 5
        LineColor='k: ';
    otherwise
        LineColor='k-'; % catchall option for anything over 5
    end

    figure(1) %set figure 1 as the active figure
    hold on % hold to stop overwriting of plots
    plot(t,h,LineColor) % plot using the character array set in the switch
statement
                                % to specify the color and line type codes
end
xlabel('Time (sec)')
ylabel('Height (m)')
title('Height as a function of Damping Coefficient & Time') % add labels and titles
ylim([-1,1]) % set height (y) limits to -1 and 1
hold off % turn off hold toggle

%%
% Create a new script and copy your CalcPosition function from Lab 2

function Height = CalcPosition(Time, Damping)
% calculating Height based on the inputs time and damping coefficient
a = 1; % defining variables for values of a and w
w = 6;
Height = exp(-Damping.*Time).*a.*cos(w.*Time); % calculate height
end

```