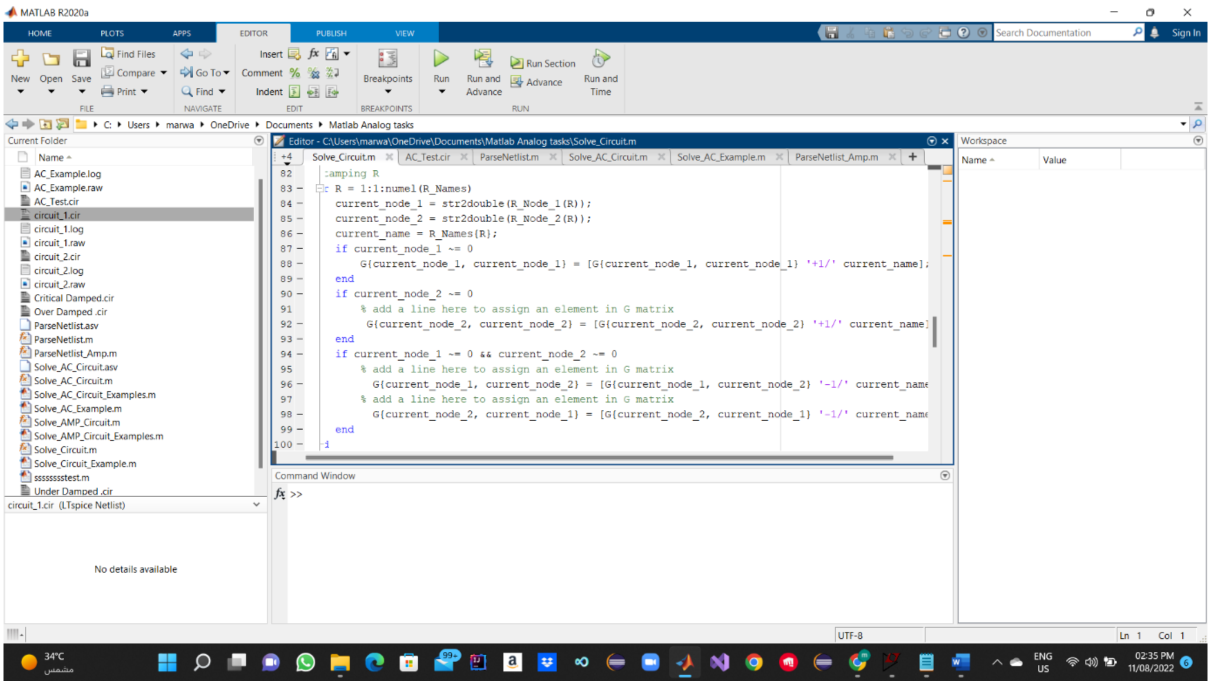
## Part 1 :

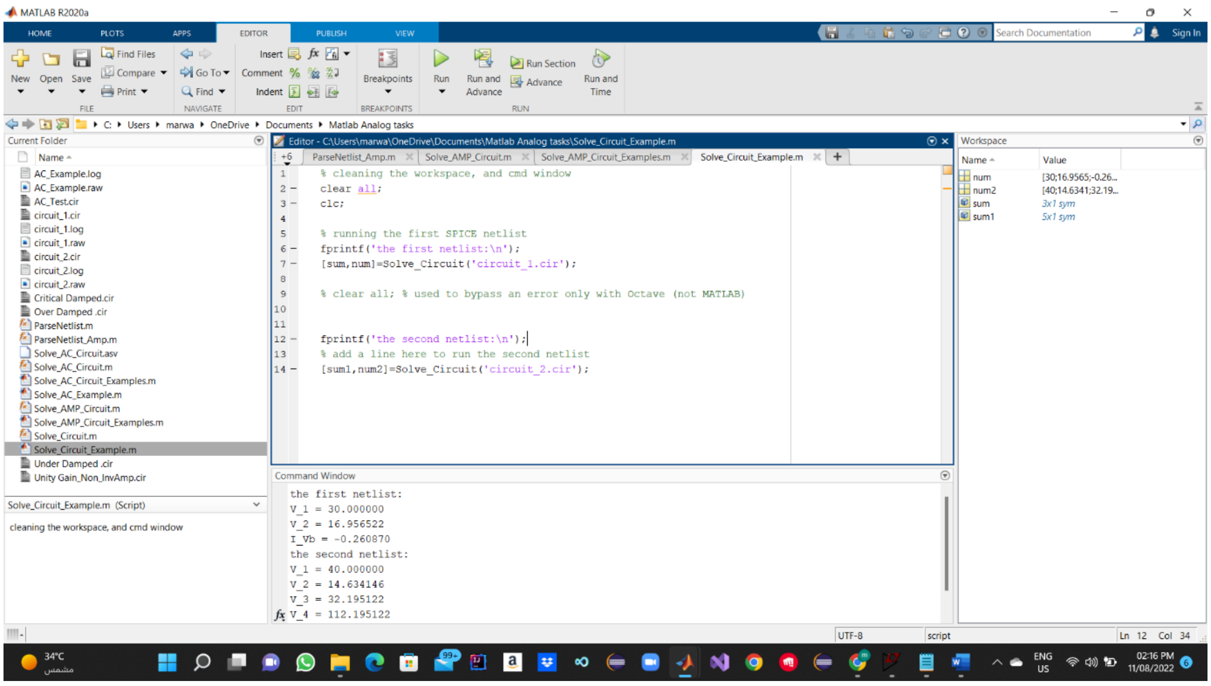
Code :



```
:amping R
r R = 1:1:numel(R_Names)
  current_node_1 = str2double(R_Node_1(R));
  current_node_2 = str2double(R_Node_2(R));
  current_name = R_Names{R};
  if current_node_1 ~= 0
      G{current_node_1, current_node_1} = [G{current_node_1, current_node_1} '+1/' current_name];
  end
  if current_node_2 ~= 0
      % add a line here to assign an element in G matrix
       G{current_node_2, current_node_2} = [G{current_node_2, current_node_2} '+1/' current_name]
  end
  if current_node_1 ~= 0 && current_node_2 ~= 0
      % add a line here to assign an element in G matrix
        G{current_node_1, current_node_2} = [G{current_node_1, current_node_2} '-1/' current_name
      % add a line here to assign an element in G matrix
        G{current_node_2, current_node_1} = [G{current_node_2, current_node_1} '-1/' current_name
  end
i
```

Simulation Results

And Solve Example is completed :



```
% cleaning the workspace, and cmd window
clear all;
clc;

% running the first SPICE netlist
fprintf('the first netlist:\n');
[sum,num]=Solve_Circuit('circuit_1.cir');

% clear all; % used to bypass an error only with Octave (not MATLAB)


fprintf('the second netlist:\n');
% add a line here to run the second netlist
[sum1,num2]=Solve_Circuit('circuit_2.cir');
```

```
the first netlist:
V_1 = 30.000000
V_2 = 16.956522
I_Vb = -0.260870
the second netlist:
V_1 = 40.000000
V_2 = 14.634146
V_3 = 32.195122
V_4 = 112.195122
```

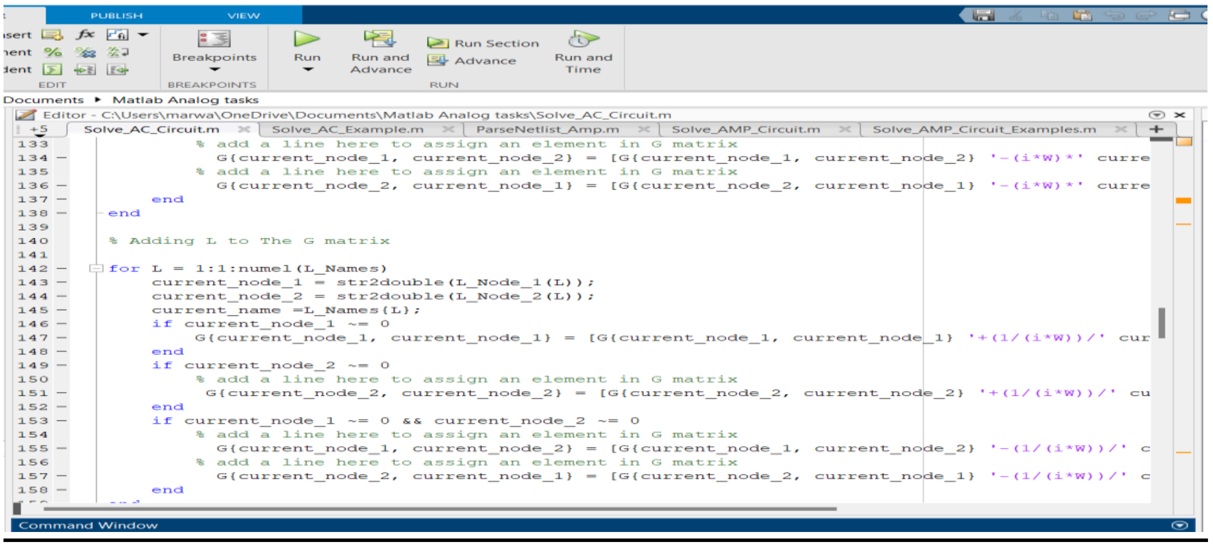
## Table Compare:

| Matlab | Ltspice |
|---|---|
| <u>the first netlist:</u><br>V_1 = 30.000000<br>V_2 = 16.956522<br>I_Vb = -0.260870<br><u>the second netlist:</u><br>V_1 = 40.000000<br>V_2 = 14.634146<br>V_3 = 32.195122<br>V_4 = 112.195122<br>I_Vb = -1.268293 | <u>the first netlist:</u><br>V(1): 30 voltage<br>V(2): 16.9565 voltage<br>I(Is):<br>2 device_current<br>I(R3): 1.69565 device_current<br>I(R2): 0.565217 device_current<br>I(R1): 0.26087 device_current<br>I(Vb): -0.26087 device_current<br><br><u>the second netlist:</u><br>**v**(1): 40 voltage<br>V(2): 14.6341 voltage<br>V(3): 32.1951 voltage<br>V(4): 112.195 voltage<br>I(Is): 1 device_current<br>I(R6): 0.804878 device_current<br>I(R4): 1.46341 device_current<br>I(R3): -1 device_current<br>I(R2): -0.195122 device_current<br>I(R1): 1.26829 device_current<br>I(Vb): -1.26829 device_current |

Ltspice Find The value of the Current in each branch while Its Not Supported in my code And The results of The voltage Nodes My Code have higher precision.

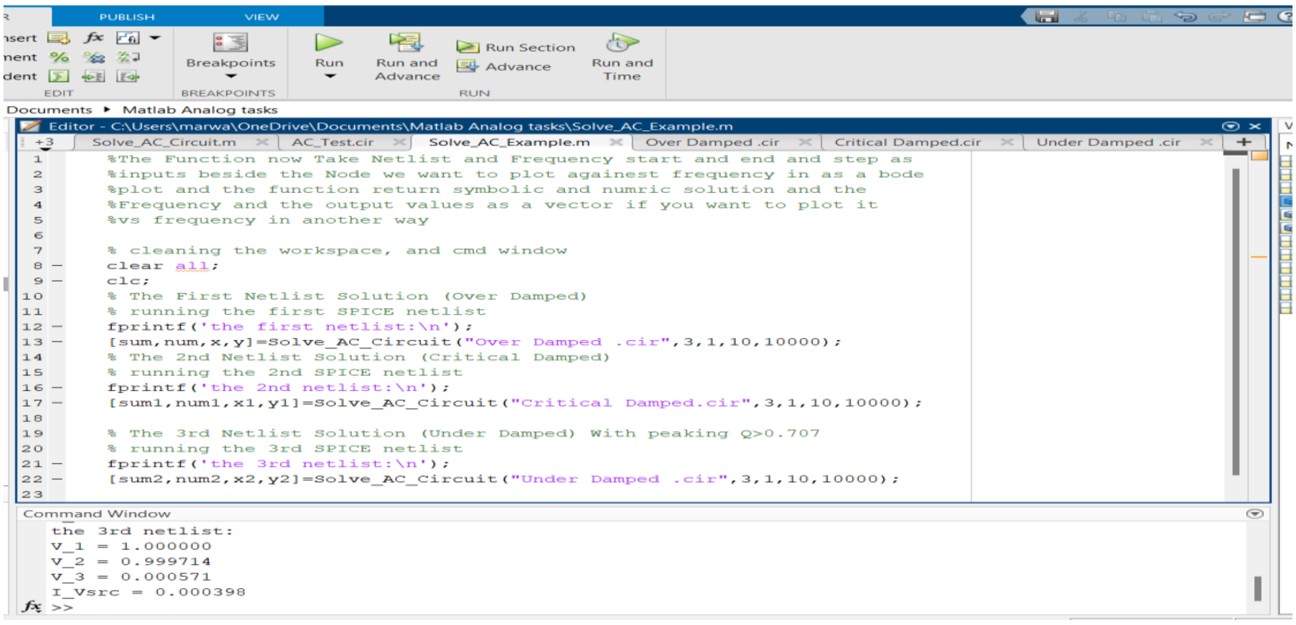
## <u>Part 2:</u>

Created the script for solving Ac.



```
                % add a line here to assign an element in G matrix
                  G{current_node_1, current_node_2} = [G{current_node_1, current_node_2} '-(i*W)*' curre
                % add a line here to assign an element in G matrix
                  G{current_node_2, current_node_1} = [G{current_node_2, current_node_1} '-(i*W)*' curre
            end
        end

        % Adding L to The G matrix

        for L = 1:1:numel(L_Names)
            current_node_1 = str2double(L_Node_1(L));
            current_node_2 = str2double(L_Node_2(L));
            current_name =L_Names{L};
            if current_node_1 ~= 0
                G{current_node_1, current_node_1} = [G{current_node_1, current_node_1} '+(1/(i*W))/' cur
            end
            if current_node_2 ~= 0
                % add a line here to assign an element in G matrix
                 G{current_node_2, current_node_2} = [G{current_node_2, current_node_2} '+(1/(i*W))/' cu
            end
            if current_node_1 ~= 0 && current_node_2 ~= 0
                % add a line here to assign an element in G matrix
                  G{current_node_1, current_node_2} = [G{current_node_1, current_node_2} '-(1/(i*W))/' c
                % add a line here to assign an element in G matrix
                  G{current_node_2, current_node_1} = [G{current_node_2, current_node_1} '-(1/(i*W))/' c
            end
```

Created the script for solving Ac Example.

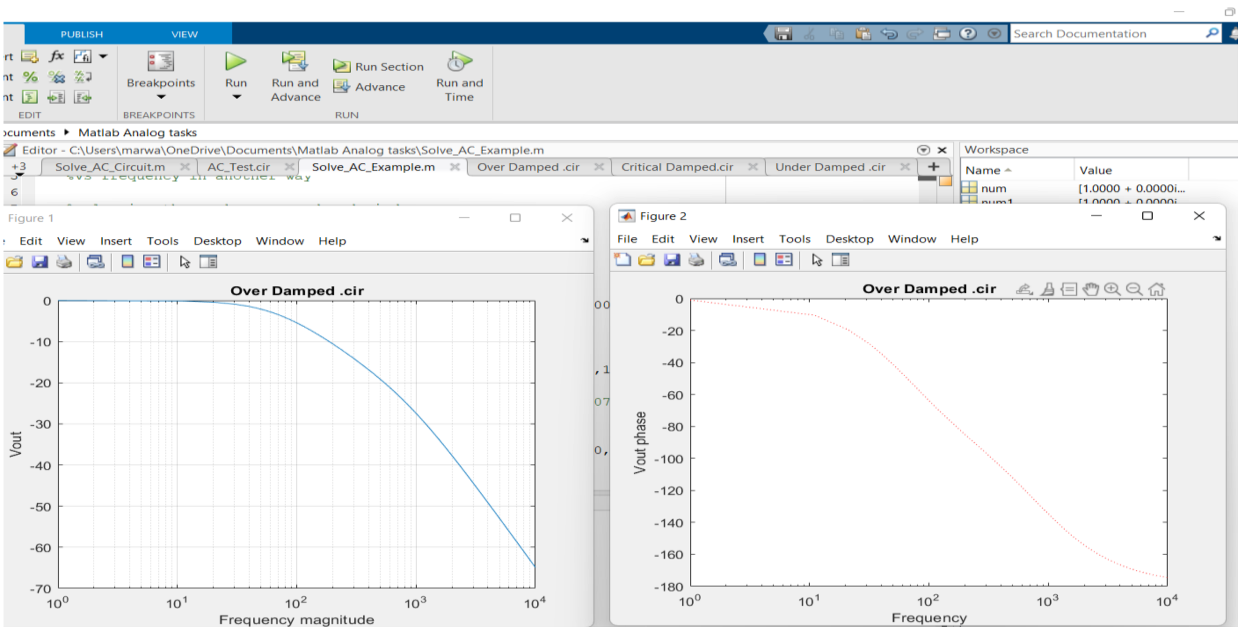


```
%The Function now Take Netlist and Frequency start and end and step as
%inputs beside the Node we want to plot againest frequency and the
%plot and the function return symbolic and numric solution and the
%Frequency and the output values as a vector if you want to plot it
%vs frequency in another way

% cleaning the workspace, and cmd window
clear all;
clc;
% The First Netlist Solution (Over Damped)
% running the first SPICE netlist
fprintf('the first netlist:\n');
[sum,num,x,y]=Solve_AC_Circuit("Over Damped .cir",3,1,10,10000);
% The 2nd Netlist Solution (Critical Damped)
% running the 2nd SPICE netlist
fprintf('the 2nd netlist:\n');
[sum1,num1,x1,y1]=Solve_AC_Circuit("Critical Damped.cir",3,1,10,10000);

% The 3rd Netlist Solution (Under Damped) With peaking Q>0.707
% running the 3rd SPICE netlist
fprintf('the 3rd netlist:\n');
[sum2,num2,x2,y2]=Solve_AC_Circuit("Under Damped .cir",3,1,10,10000);
```

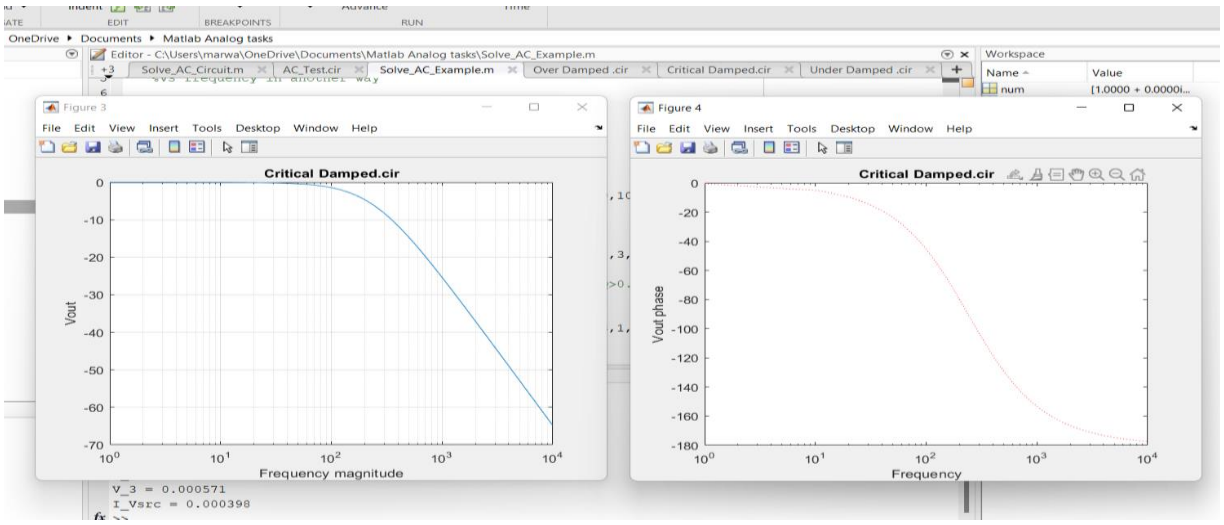
```
the 3rd netlist:
V_1 = 1.000000
V_2 = 0.999714
V_3 = 0.000571
I_Vsrc = 0.000398
>>
```

## My Simulator Results:

### Over Damped RLC:

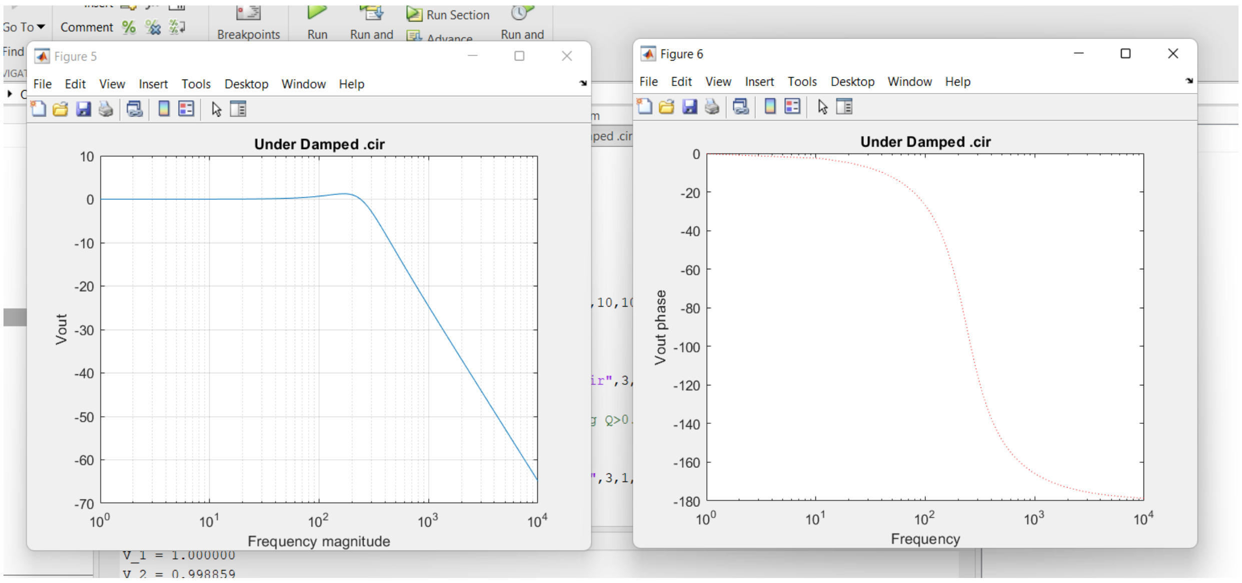


## Critical Damped RLC:



```
V_3 = 0.000571
I_Vsrc = 0.000398
```

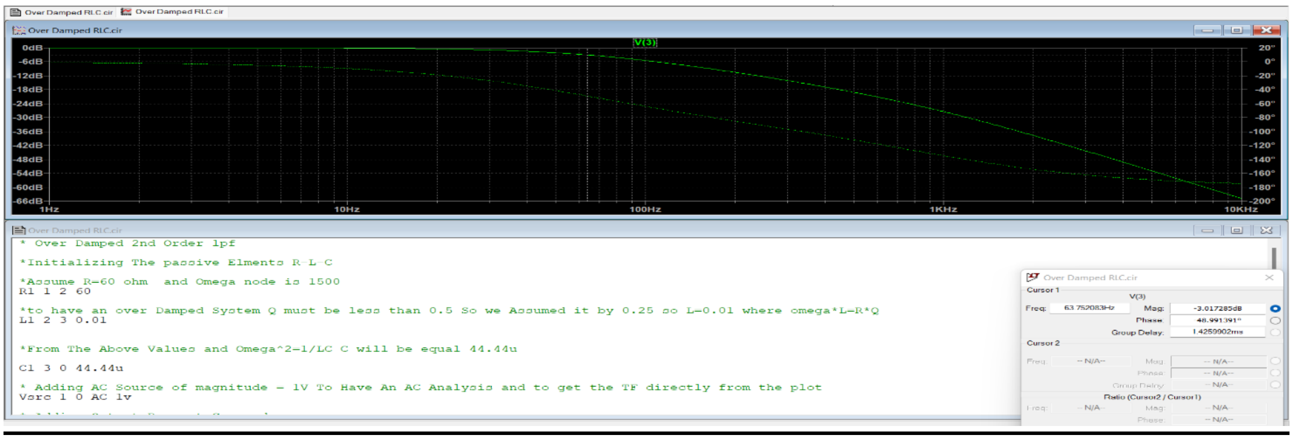
## Under Damped RLC With Peaking :



```
V_1 = 1.000000
V_2 = 0.998859
```

## LTspice Results:

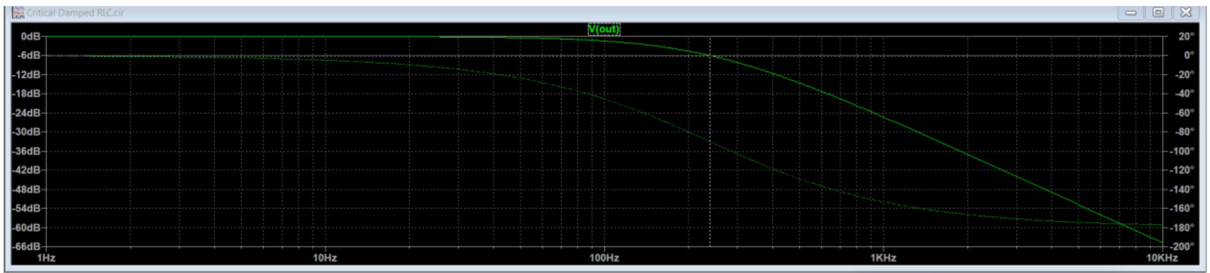
### Over Damped RLC:

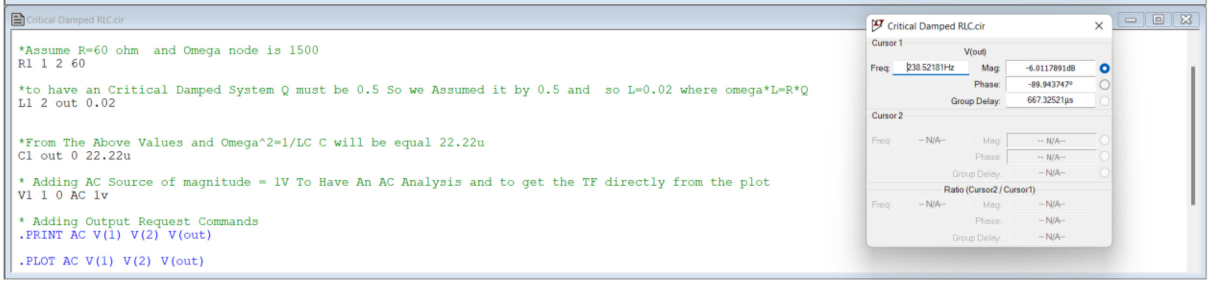


```
* Over Damped 2nd Order lpf

*Initializing The passive Elments R-L-C

*Assume R=60 ohm  and Omega node is 1500
R1 1 2 60

*to have an over Damped System Q must be less than 0.5 So we Assumed it by 0.25 so L=0.01 where omega*L=R*Q
L1 2 3 0.01


*From The Above Values and Omega^2=1/LC C will be equal 44.44u

C1 3 0 44.44u

* Adding AC Source of magnitude = 1V To Have An AC Analysis and to get the TF directly from the plot
Vsrc 1 0 AC 1v
```

Over Damped RLC.cir
Cursor 1
V(3)
Freq: 63.752083Hz   Mag: -3.017285dB
Phase: -48.991391°
Group Delay: 1.4259902ms
Cursor 2
Freq: -- N/A--   Mag: -- N/A--
Phase: -- N/A--
Group Delay: -- N/A--
Ratio (Cursor2 / Cursor1)
Freq: -- N/A--   Mag: -- N/A--
Phase: -- N/A--

Critical Damping RLC:

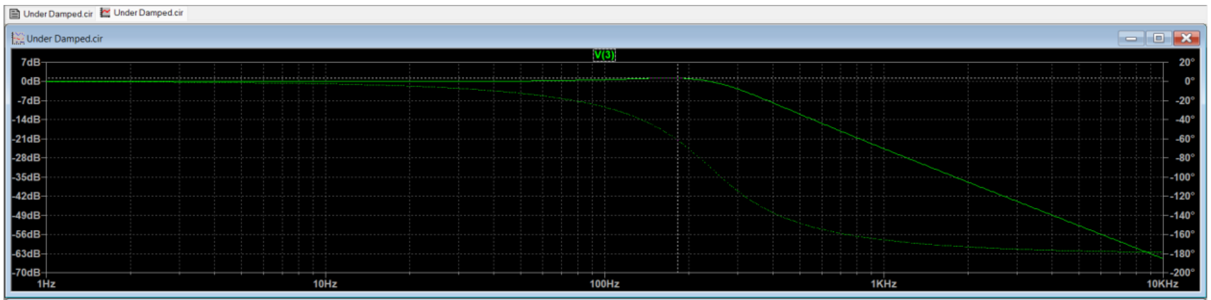


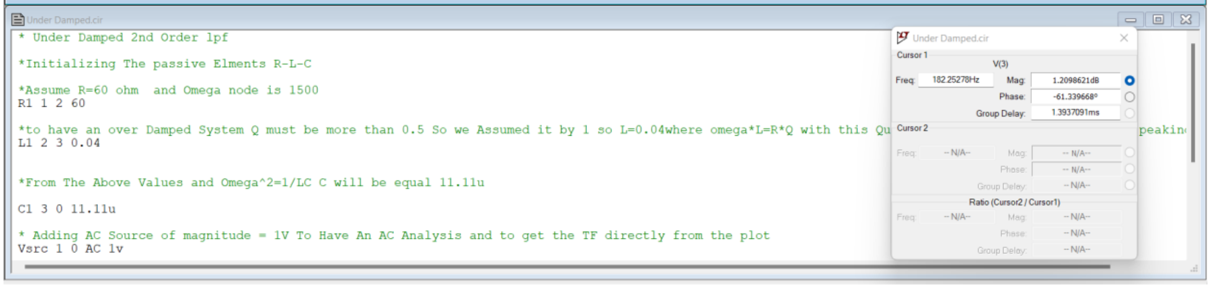


```
*Assume R=60 ohm  and Omega node is 1500
R1 1 2 60

*to have an Critical Damped System Q must be 0.5 So we Assumed it by 0.5 and  so L=0.02 where omega*L=R*Q
L1 2 out 0.02


*From The Above Values and Omega^2=1/LC C will be equal 22.22u
C1 out 0 22.22u

* Adding AC Source of magnitude = 1V To Have An AC Analysis and to get the TF directly from the plot
V1 1 0 AC 1v

* Adding Output Request Commands
.PRINT AC V(1) V(2) V(out)

.PLOT AC V(1) V(2) V(out)
```

```
Critical Damped RLC.cir
Cursor 1
                     V(out)
Freq: 238.52181Hz    Mag:          -6.0117891dB
                     Phase:        -89.943747°
                     Group Delay:  667.32521µs
Cursor 2
Freq: -- N/A--       Mag:          -- N/A--
                     Phase:        -- N/A--
                     Group Delay:  -- N/A--
              Ratio (Cursor2 / Cursor1)
Freq: -- N/A--       Mag:          -- N/A--
                     Phase:        -- N/A--
                     Group Delay:  -- N/A--
```

Under Damped RLC:





```
* Under Damped 2nd Order lpf

*Initializing The passive Elments R-L-C

*Assume R=60 ohm  and Omega node is 1500
R1 1 2 60

*to have an over Damped System Q must be more than 0.5 So we Assumed it by 1 so L=0.04where omega*L=R*Q with this Qu                                  peakin
L1 2 3 0.04


*From The Above Values and Omega^2=1/LC C will be equal 11.11u

C1 3 0 11.11u

* Adding AC Source of magnitude = 1V To Have An AC Analysis and to get the TF directly from the plot
Vsrc 1 0 AC 1v
```

```
Under Damped.cir
Cursor 1
                     V(3)
Freq: 182.25278Hz    Mag:          1.2098621dB
                     Phase:        -61.339668°
                     Group Delay:  1.3937091ms
Cursor 2
Freq: -- N/A--       Mag:          -- N/A--
                     Phase:        -- N/A--
                     Group Delay:  -- N/A--
              Ratio (Cursor2 / Cursor1)
Freq: -- N/A--       Mag:          -- N/A--
                     Phase:        -- N/A--
                     Group Delay:  -- N/A--
```

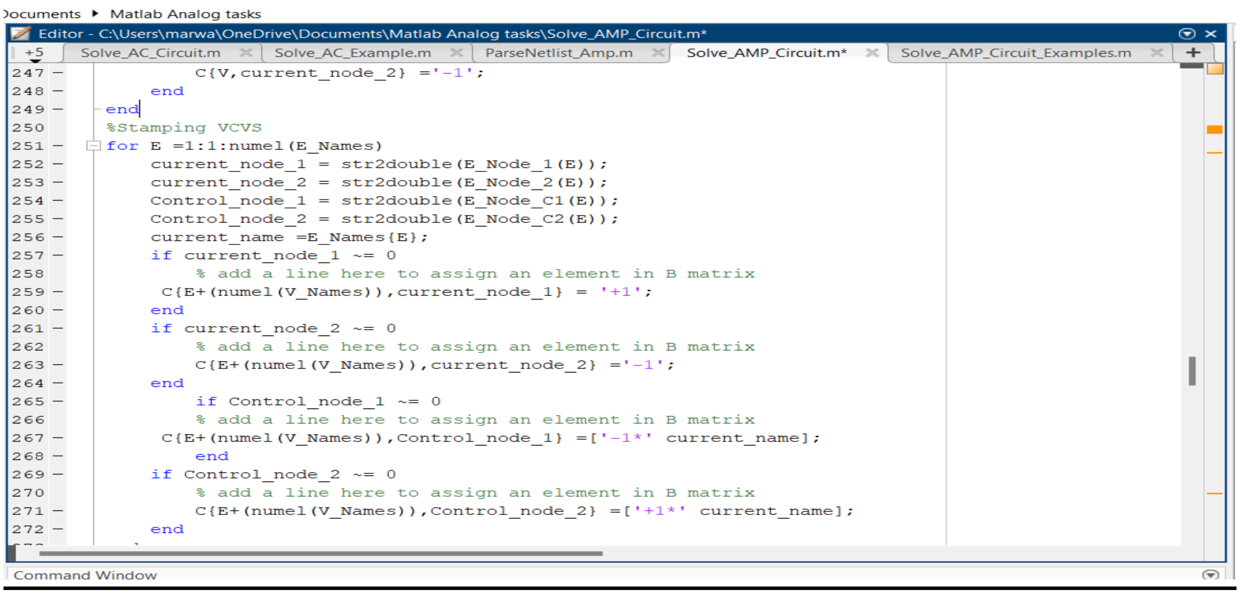
## Comment:

The Graphs Are The Same For Both Lt Spice and my Simulator

But LtSpice Is faster at getting Results and have more helping tools like cursor to have an accurate number for the cutoff Freq etc.

## PART 3:

Added Support for both VCVS and VCCS :

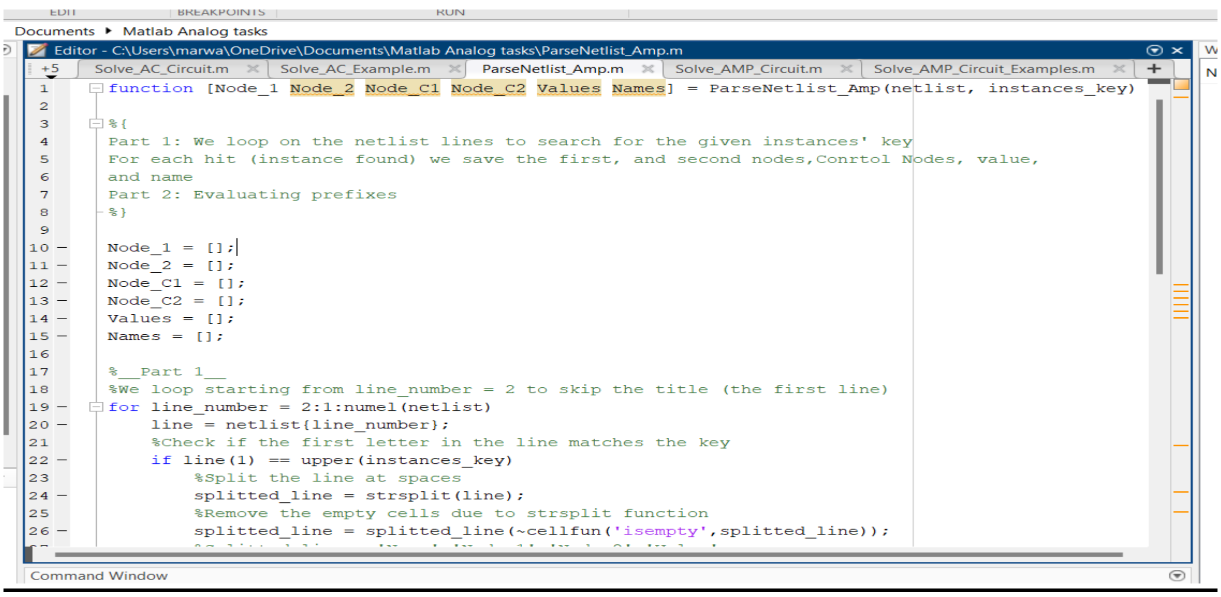


```
            C{V,current_node_2} ='-1';
        end
    end
    %Stamping VCVS
    for E =1:1:numel(E_Names)
        current_node_1 = str2double(E_Node_1(E));
        current_node_2 = str2double(E_Node_2(E));
        Control_node_1 = str2double(E_Node_C1(E));
        Control_node_2 = str2double(E_Node_C2(E));
        current_name =E_Names{E};
        if current_node_1 ~= 0
            % add a line here to assign an element in B matrix
         C{E+(numel(V_Names)),current_node_1} = '+1';
        end
        if current_node_2 ~= 0
            % add a line here to assign an element in B matrix
            C{E+(numel(V_Names)),current_node_2} ='-1';
        end
            if Control_node_1 ~= 0
            % add a line here to assign an element in B matrix
         C{E+(numel(V_Names)),Control_node_1} =['-1*' current_name];
            end
        if Control_node_2 ~= 0
            % add a line here to assign an element in B matrix
            C{E+(numel(V_Names)),Control_node_2} =['+1*' current_name];
        end
```

## Note:

Modified The Parseing function to add support for VCVS And VCCS

I used the modifeid one along with the original one ,its not optimum for  coding it can be done with the orginal one with some modifications only.



```
function [Node_1 Node_2 Node_C1 Node_C2 Values Names] = ParseNetlist_Amp(netlist, instances_key)

%{
Part 1: We loop on the netlist lines to search for the given instances' key
For each hit (instance found) we save the first, and second nodes,Conrtol Nodes, value,
and name
Part 2: Evaluating prefixes
%}

Node_1 = [];
Node_2 = [];
Node_C1 = [];
Node_C2 = [];
Values = [];
Names = [];

%__Part 1__
%We loop starting from line_number = 2 to skip the title (the first line)
for line_number = 2:1:numel(netlist)
    line = netlist{line_number};
    %Check if the first letter in the line matches the key
    if line(1) == upper(instances_key)
        %Split the line at spaces
        splitted_line = strsplit(line);
        %Remove the empty cells due to strsplit function
        splitted_line = splitted_line(~cellfun('isempty',splitted_line));
```

## My Simulator Results :

Non Inverting amplifier with UGF=10 Megahz connected in UGain Configuration

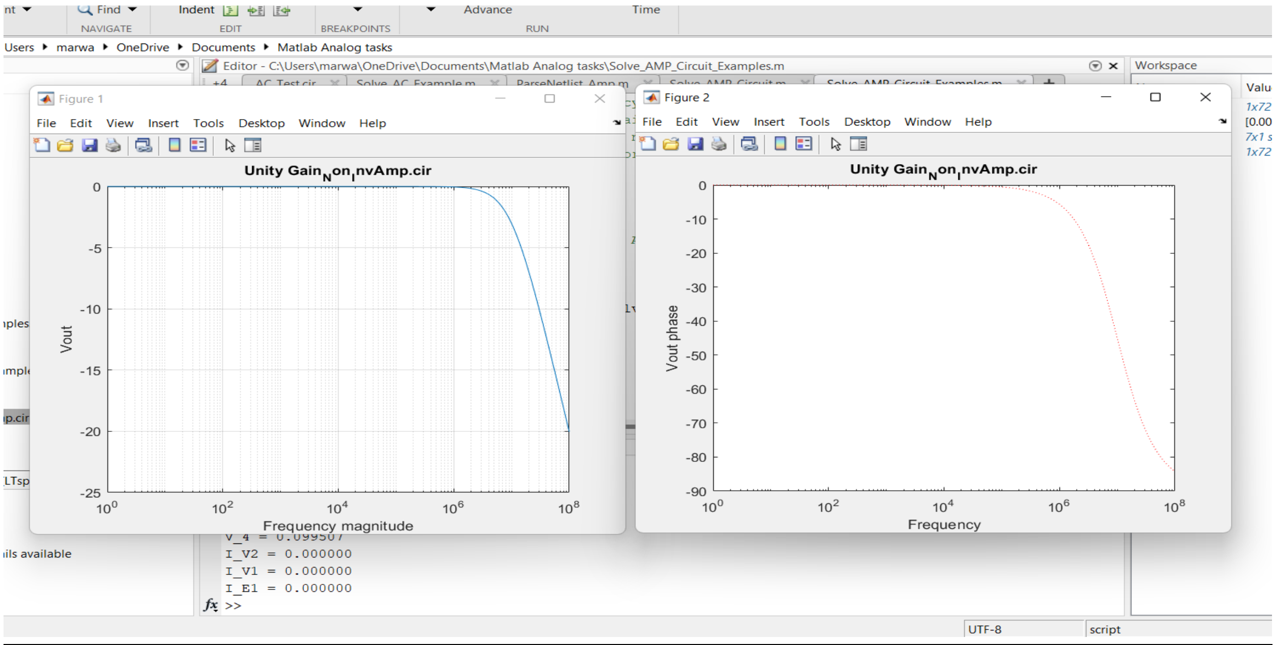
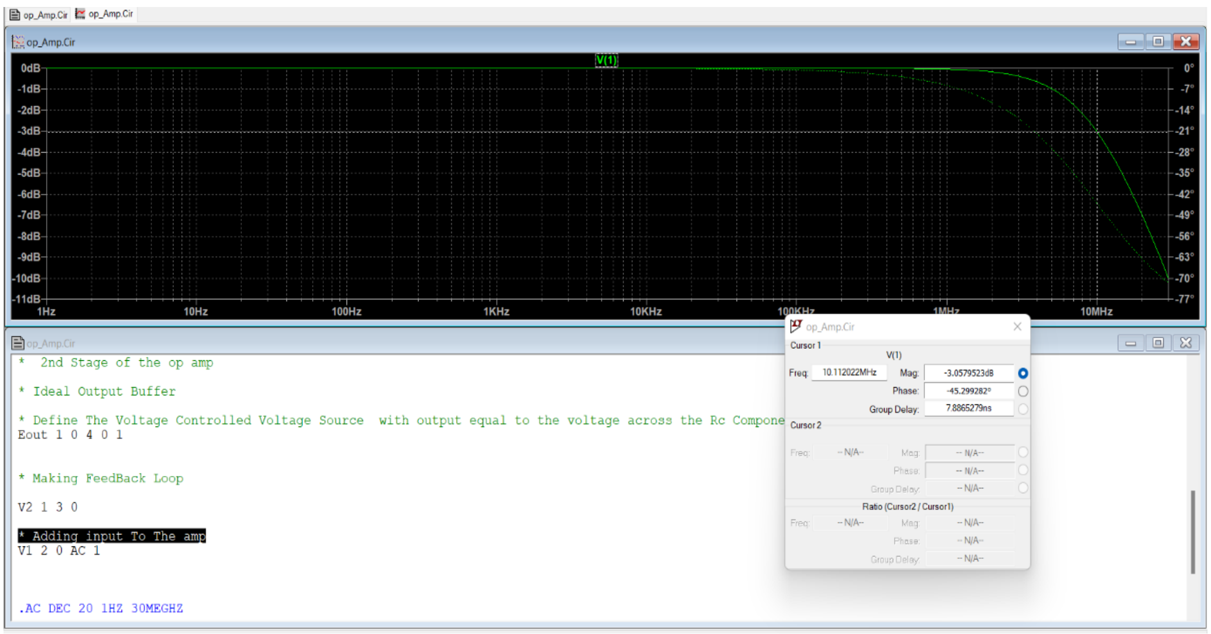


Figure 1
Unity Gain_Non_InvAmp.cir
Vout
Frequency magnitude
Figure 2
Unity Gain_Non_InvAmp.cir
Vout phase
Frequency
Editor - C:\Users\marwa\OneDrive\Documents\Matlab Analog tasks\Solve_AMP_Circuit_Examples.m

```
V_4 = 0.099507
I_V2 = 0.000000
I_V1 = 0.000000
I_E1 = 0.000000
>>
```

## Lt SPice Results :



```
*  2nd Stage of the op amp

* Ideal Output Buffer

* Define The Voltage Controlled Voltage Source  with output equal to the voltage across the Rc Compone
Eout 1 0 4 0 1

* Making FeedBack Loop

V2 1 3 0

* Adding input To The amp
V1 2 0 AC 1

.AC DEC 20 1HZ 30MEGHZ
```

Cursor 1 — V(1)

Freq: 10.112022MHz  Mag: -3.0579523dB
Phase: -45.299282°
Group Delay: 7.8865279ns

Cursor 2

Freq: -- N/A--  Mag: -- N/A--
Phase: -- N/A--
Group Delay: -- N/A--

Ratio (Cursor2 / Cursor1)

Freq: -- N/A--  Mag: -- N/A--
Phase: -- N/A--
Group Delay: -- N/A--

Comment: The Graphs are Identical The Simulator is spice accurate but its much slower than spice.