MANDATORY TASK

Task 1

In this task, you create a simple circuit simulator in MATLAB. Assume that an arbitrary in-rest LTI RLC circuit is given. The circuit is characterized by a structure that describes the circuit elements. For example, for the sample circuit of Fig. 1, the describing structure is as follows. Note that the circuit topology can be extracted from the describing structure. Further, remember that the values of the element as well as the phasor of the independent sources are given by the "value" field of the structure. For more information, read the comments accompanied by the sample structure.

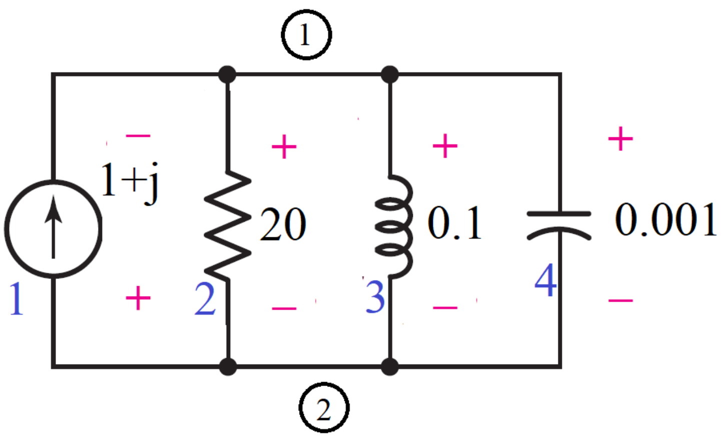


Figure 1: A sample LTI RLC circuit. The node IDs are given is circle while the element IDs are shown in blue.

```
% current source
element(1).id = 1; % element ID number
element(1).posnode = 2; % node ID connected to the element positive terminal
element(1).negnode = 1; % node ID connected to the element negative terminal
element(1).type = 'I'; % can be R (resistor), C (capacitor), L (inductor), V (independent
    voltage source), I (independent current source)
```

Task 1 continued on next page...

```
element(1).value = 1+1i; % value is a real number for R, L, and C elements and is a complex phasor for independent sources
element(1).volres = []; % a complex vector which is filled by element voltage frequency response
element(1).curres = []; % a complex vector which is filled by element current frequency response

% resistor
element(2).id = 2; % element ID number
element(2).posnode = 1; % node ID connected to the element positive terminal
element(2).negnode = 2; % node ID connected to the element negative terminal
element(2).type = 'R'; % can be R (resistor), C (capacitor), L (inductor), V (independent voltage source), I (independent current source)
element(2).value = 20; % value is a real number for R, L, and C elements and is a complex phasor for independent sources
element(2).volres = []; % a complex vector which is filled by element voltage frequency response
element(2).curres = []; % a complex vector which is filled by element current frequency response

% inductor
element(3).id = 3; % element ID number
element(3).posnode = 1; % node ID connected to the element positive terminal
element(3).negnode = 2; % node ID connected to the element negative terminal
element(3).type = 'L'; % can be R (resistor), C (capacitor), L (inductor), V (independent voltage source), I (independent current source)
element(3).value = 0.1; % value is a real number for R, L, and C elements and is a complex phasor for independent sources
element(3).volres = []; % a complex vector which is filled by element voltage frequency response
element(3).curres = []; % a complex vector which is filled by element current frequency response

% capacitor
element(4).id = 4; % element ID number
element(4).posnode = 1; % node ID connected to the element positive terminal
element(4).negnode = 2; % node ID connected to the element negative terminal
element(4).type = 'C'; % can be R (resistor), C (capacitor), L (inductor), V (independent voltage source), I (independent current source)
element(4).value = 0.001; % value is a real number for R, L, and C elements and is a complex phasor for independent sources
element(4).volres = []; % a complex vector which is filled by element voltage frequency response
```

Task 1 continued on next page...

```
element(4).curres = []; % a complex vector which is filled by element current frequency
    response
```

(a) Write a MATLAB function as "[element] = simulator(element, frequency)" that performs AC analysis for a given in-rest LTI RLC circuit over a desired positive frequency range such as $1 : 1 : 1000$. In fact, the function gets the element structure and frequency range as its inputs, computes the current and voltage frequency responses of the elements, fills the empty response fields of the element structure, and returns the filed element structure. You might use node, mesh, cut-set, or loop analysis for the function implementation. For this part, assume that the input frequency range covers a positive interval and the circuit has no problematic alone independent sources.

(b) Now, let the input frequency range include 0. Particularly, it may only be a single $0 : 1 : 0$ value denoting the DC analysis. Improve your implementation to cover the DC analysis.

(c) Now, extend the function to handle the existence of the problematic independent sources.

(d) Use the developed simulator to analyze the frequency response of the given sample network in Fig. 1. Use the filled element structure to draw the voltage frequency response of the circuit. Plot the corresponding bode diagrams,

(e) Examine your developed simulator for several other circuits and discuss the obtained results.

(f) What does happen for your simulator if an exceptional case violating the KCL or KVL rules, such as two parallel voltage sources with different voltage values, occur in the circuit.

(g) Prepare a short report and describe your work concisely. Use suitable figures to better describe the developed codes and to make your report more readable and understandable. Attach a copy of the developed codes to your sent report. Make sure to provide descriptive comments for the codes.

BONUS

Task 2

You can get extra score by doing the optional tasks listed below.

(a) Extend the developed function to cover the existence of linear dependent sources. Plot the frequency response of a circuit having dependent sources using the developed simulator.

(b) Extend the developed function to cover the existence of coupled linear inductors and transformers. Plot the frequency response of a coupled circuit using the developed simulator.

(c) Improve the implementation to cover the transient analysis. You may need to add new fields to the element structure to include initial conditions and time profiles of independent sources.

(d) Return your report by filling the LaTeXtemplate of the project. If you want to add a circuit schematic or a diagram, you can draw it directly using TikZ package, or draw it in a secondary application such as Microsoft Visio and then, import it as a figure. Another option is the website

REPORT

At first the type of analysis must be determined to be **AC** or **DC** or **Transient**. After selecting the type of analysis, it receives the required information.
Then we enter the number of elements and receive the information of each element separately. In this loop, some required variables are set by default by entering information and some characteristics are determined by entering information, such as impedance, which is calculated with the **getAmpedance** function and if the information is entered completely, the following message will be received.

" THE INFORMATION OF THIS ELEMENT WAS COMPLETED. "

To check the code, test cases are designed, each of which is dedicated to checking a specific task. for example, coupled inductors and dependent sources, etc. Now, by determining the number of nodes and branches, we perform the simulation using the simulator function. If the **KCL** and **KVL** are not violated and the simulation is not stopped, this function will output the voltage and current of all branches.Otherwise, if **KCL** and **KVL** are violated, the following outputs are obtained.

" KVL/KCL has been violated!! Please correct the circuit. "

How the simulator function works :

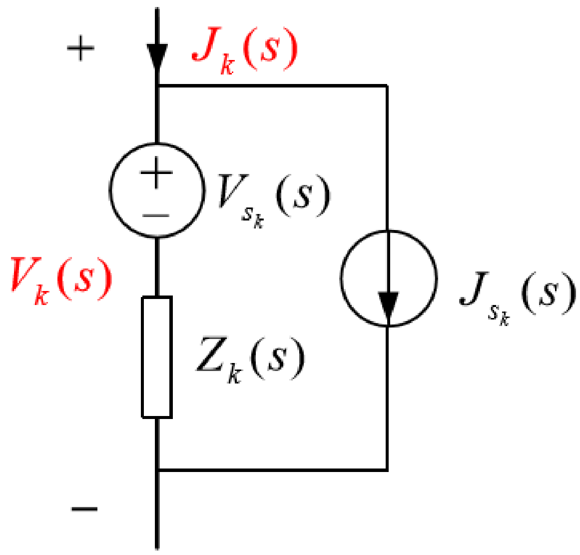
First check that **KCL** and **KVL** are not violated.

```
for i=1:element_number
    for j=1:element_number
        if ~strcmp(element(j).type , 'I')
            all_isource = 0;
            break;
        else
            all_isource = 1;
        end
    end
    for j=1:element_number
        if (j ~= i) && (element(j).posnode == element(i).posnode)
            in_one_direction = 0;
        end
    end
    for j=1:element_number
        if (j ~= i) && (element(j).value ~= element(i).value)
            all_equal = 0;
        end
    end
end
if all_isource && in_one_direction && all_equal
    KCL_violation = 0;
    stopsim = 1;
    disp('The solution of this cicrcuit is not unique! (Series equal current sources)');
elseif all_isource
    KCL_violation = 1;
    stopsim = 1;
    disp('KCL has been violated!! Please correct the circuit.');
end
```

```
for i=1:element_number
        for j=1:element_number
            if ~strcmpi(element(j).type , 'V')
                all_vsource = 0;
                break;
            else
                all_vsource = 1;
            end
        end
    for j=1:element_number
        if (j ~= i) && (element(j).posnode == element(i).negnode)
            same_polarity = 0;
        end
    end
    for j=1:element_number
        if (j ~= i) && (element(j).value ~= element(i).value)
            all_equal = 0;
        end
    end
    end
    if all_vsource && same_polarity && all_equal
        KVL_violation = 0;
        stopsim = 1;
        disp('The solution of this cicrcuit is not unique! (Parallel equal voltage sources)');
    elseif all_vsource
        KVL_violation = 1;
        stopsim = 1;
        disp('KVL has been violated!! Please correct the circuit.');
    end
end
```

Now we code according to the following theory.

We consider each element according to the following figure.



We create matrix **A** according to the following description.

$$A_a = [a_{ik}]_{n_t \times b}$$

$$a_{ik} = \begin{cases} 1 & \text{if branch } k \text{ } \textit{leaves} \text{ node } (i) \\ -1 & \text{if branch } k \text{ } \textit{enters} \text{ node } (i) \\ 0 & \text{if branch } k \text{ is } \textit{not incident with} \text{ node } (i) \end{cases}$$

ماتریس تلاقی گره با شاخه

– تعداد سطرها برابر تعداد گره‌ها
– تعداد ستون‌ها برابر تعداد شاخه‌ها
– هر ستون فقط یک عدد ۱ و یک عدد ۱− دارد

By forming an **A** matrix, **KCL** and **KVL** can be written.

$$KCL : AJ = 0$$
$$KVL : V = A^T E$$

Then we create the **b** $\times$ **b** square matrix Y_b in such a way that the main diameter of the matrix is the admittance of the branches and the other entry of a matrix be zero but in the presence of coupled inductors or dependent sources, this matrix is out of diameter.

Now it's time to form the Y_n matrix.

$$Y_n = AY_bA^T$$

Then we create two matrices V_s and J_s by using the **create Vs** and **create Js** functions. V_s is a matrix that filled with the V_s attribute of each element. V_s all elements are zero unless we have a voltage source, in which case the value of this characteristic changes. J_s is a matrix that filled with the J_s attribute of each element. J_s all elements are zero unless we have a voltage source, in which case the value of this characteristic changes. With J_s and V_s we can create I_s with using this formula.

$$I_s = A(Y_b V_s - J_s)$$

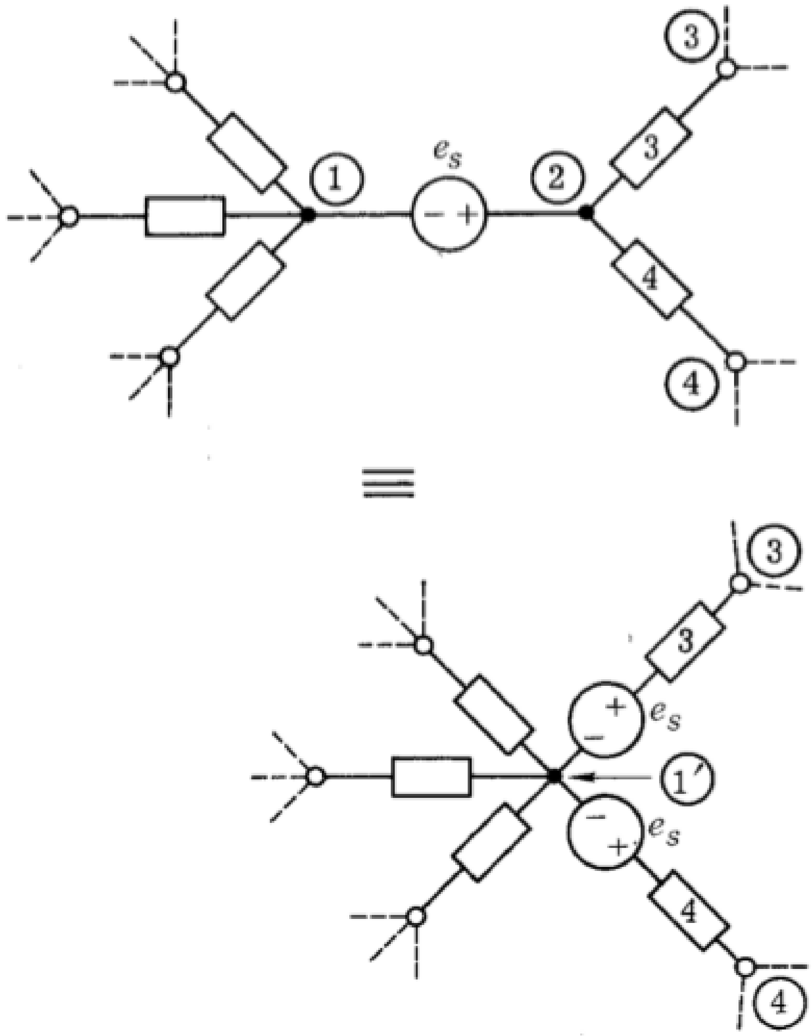
NOw we can calculate E_s with having Y_n and I_s

$$Y_n E_s = I_s$$

Now, according to two formulas of below, the voltage and current of all branches are calculated.

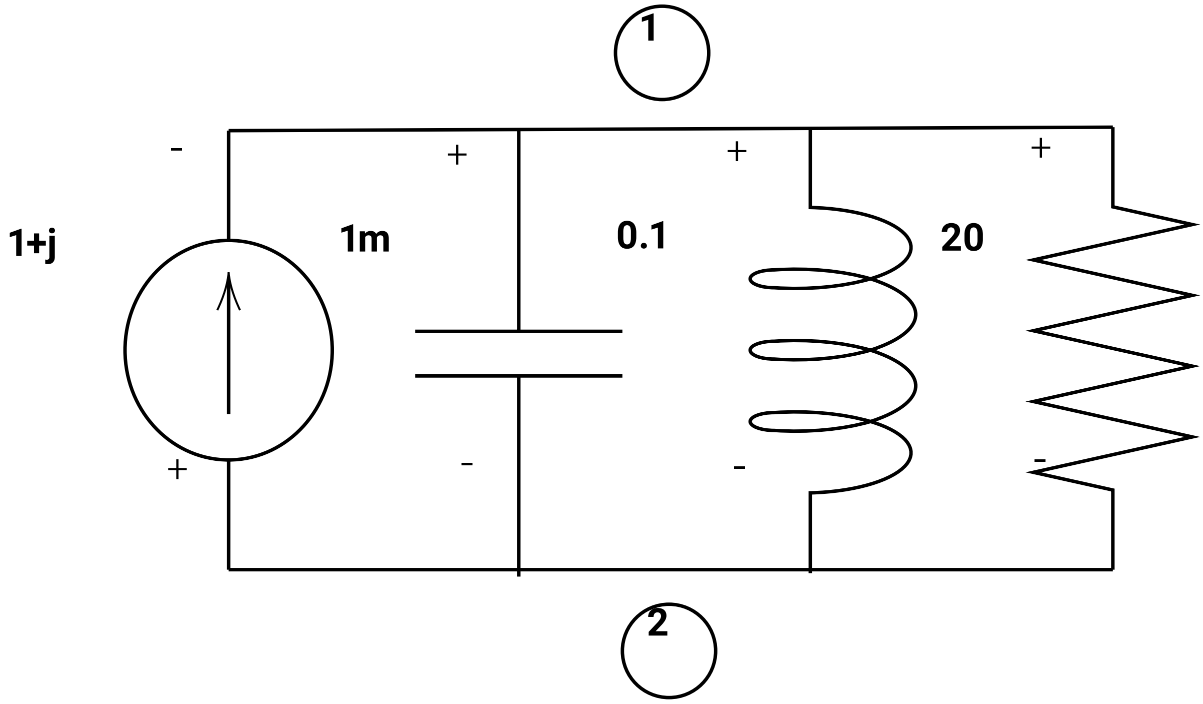
$$V = A^T E_s$$
$$J_s = Y_b V + J_s - Y_b V_s$$

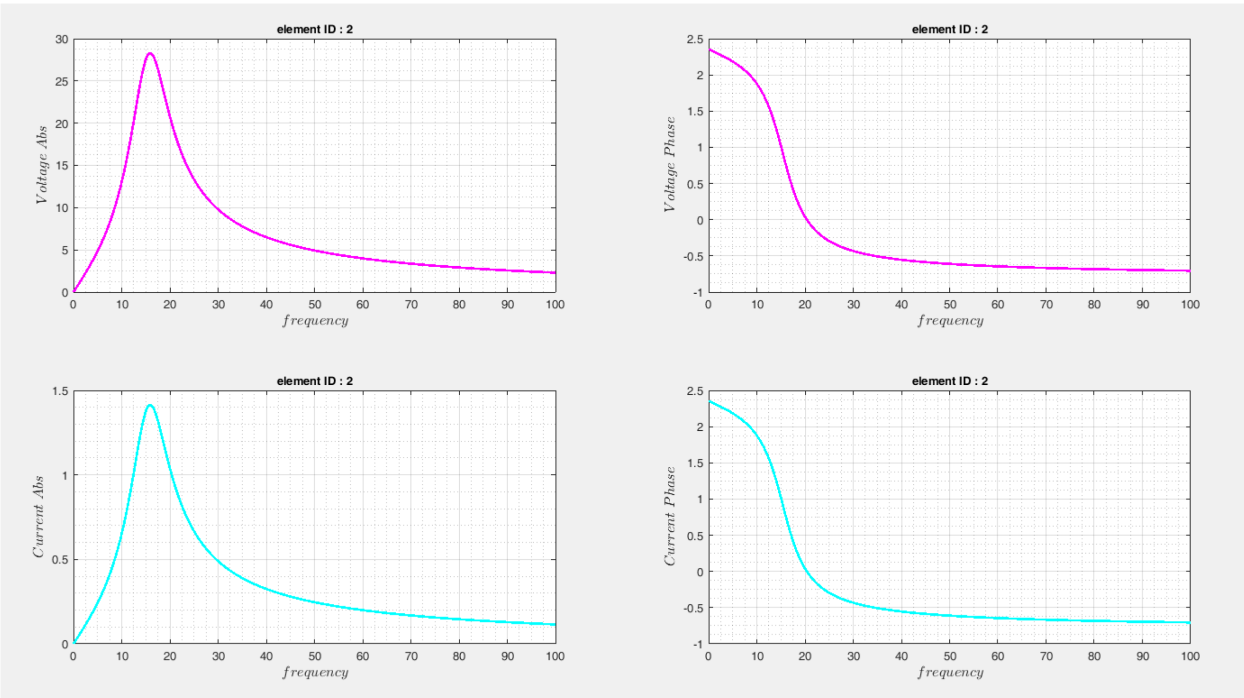
The **simulator** function also handles annoying sources (in the node method : voltage source) using the **ProblematicSource** function. This function solves the problem by converting the voltage source (transfer to the branches connected to one of the nodes in series and short-circuit that branch, which in fact removes that branch according to the figure below and reduces the number of nodes by one).

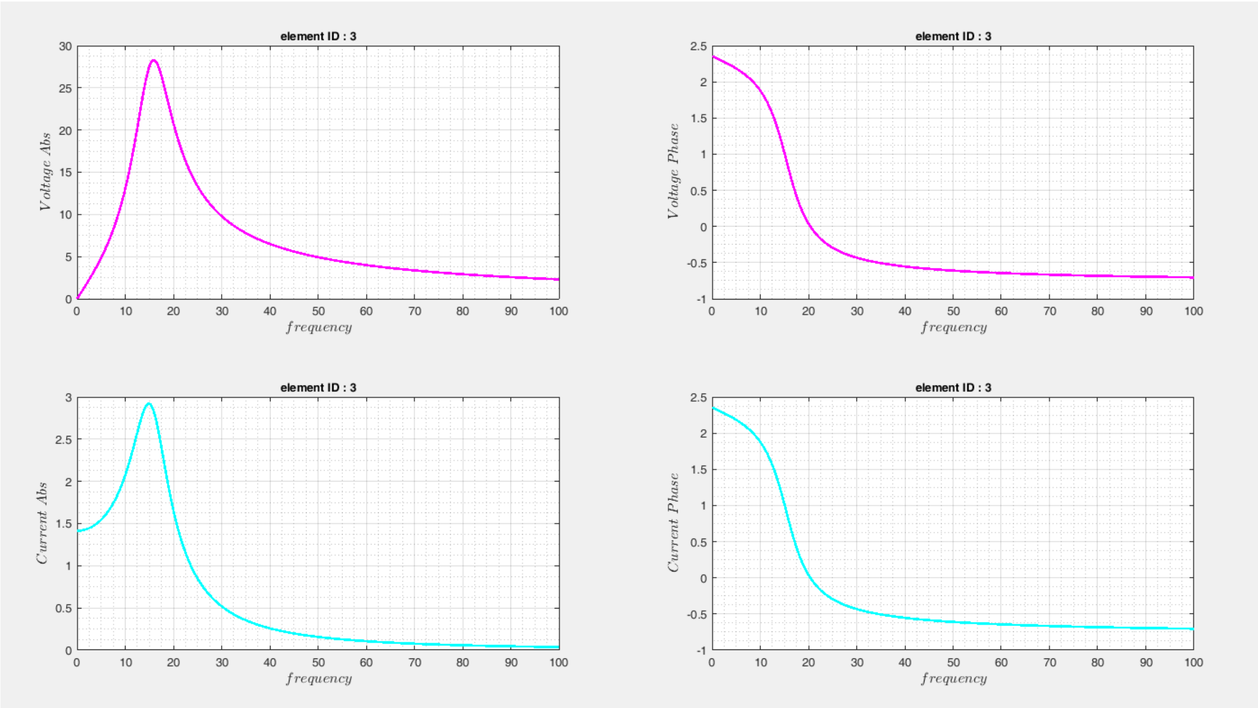

e_s
1
2
3
3
4
4
e_s
e_s
3
3
1'
4
4

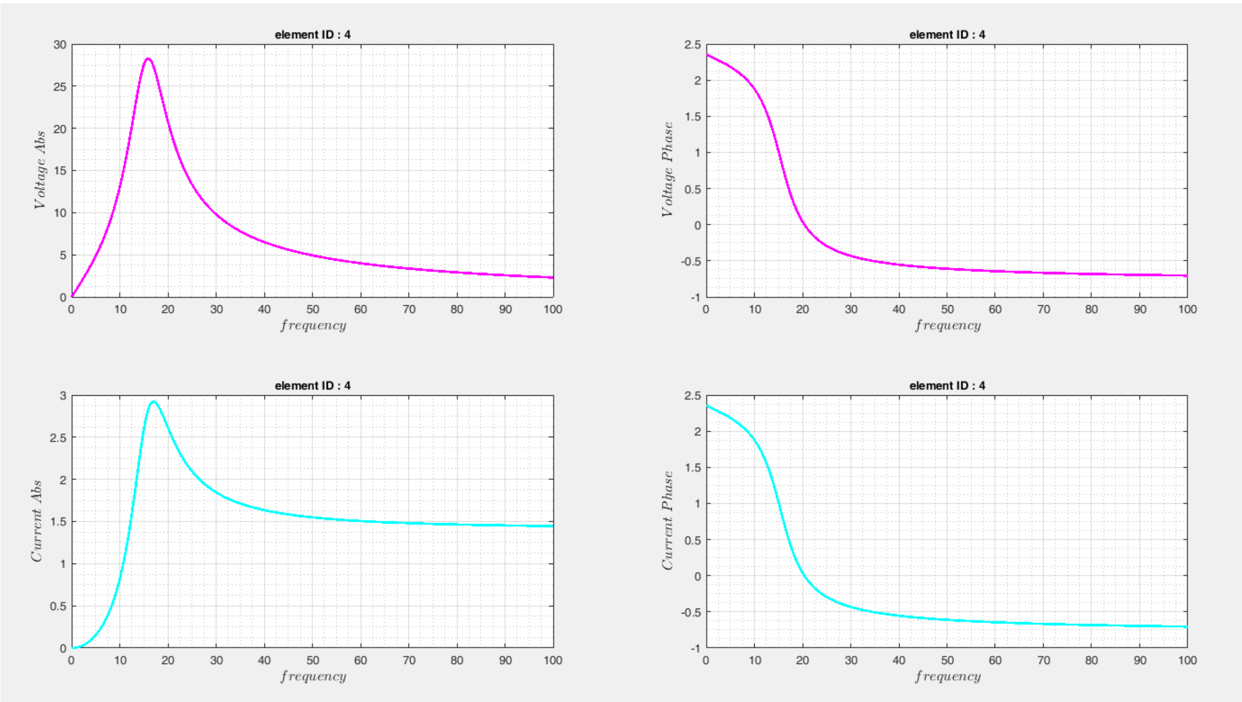
TESTCASES

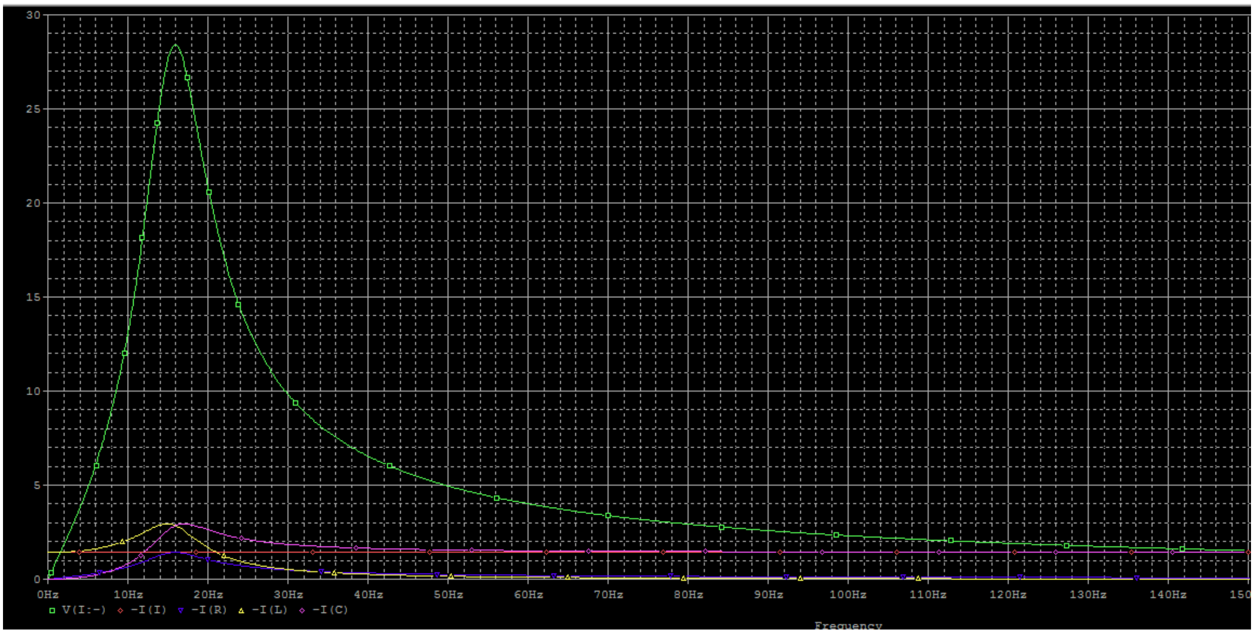
RLC Parallel



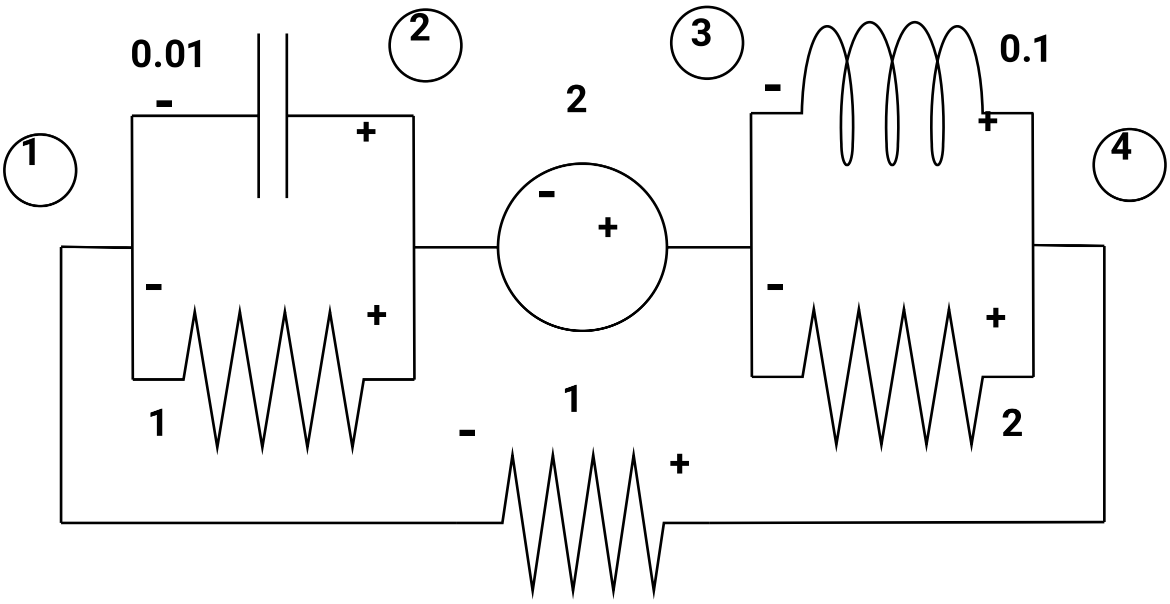


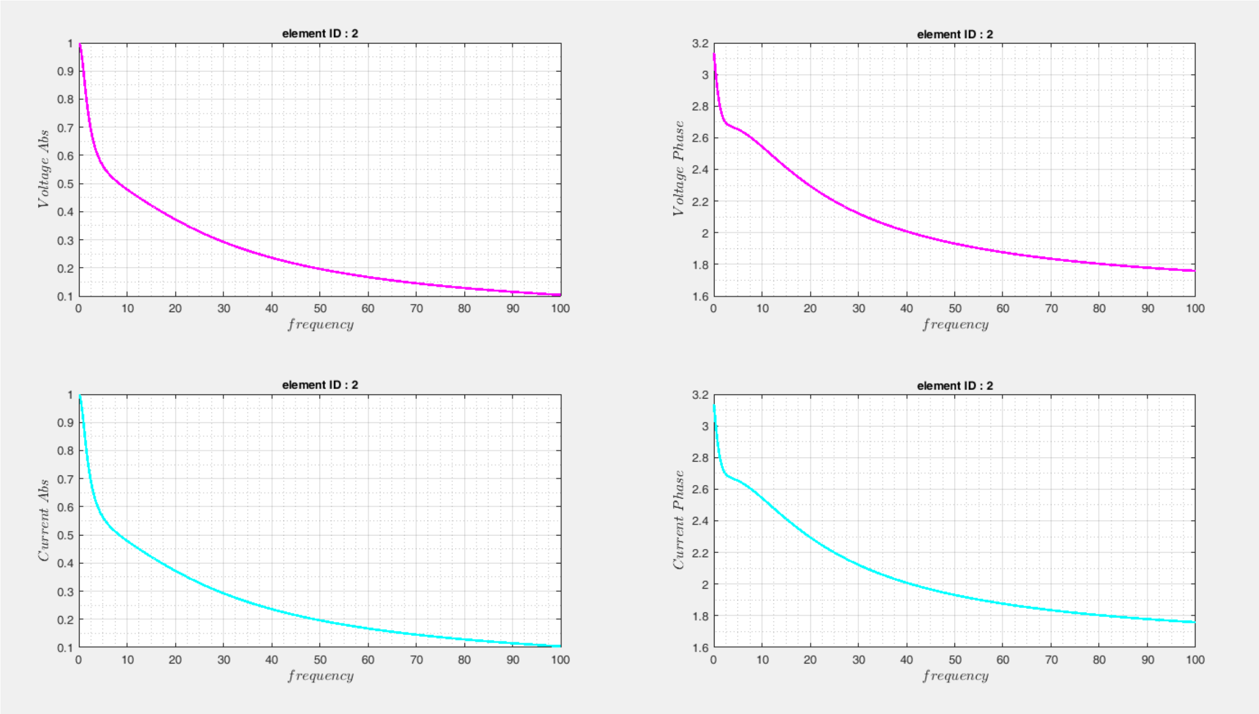

element ID : 3
Voltage Abs
frequency
element ID : 3
Voltage Phase
frequency
element ID : 3
Current Abs
frequency
element ID : 3
Current Phase
frequency

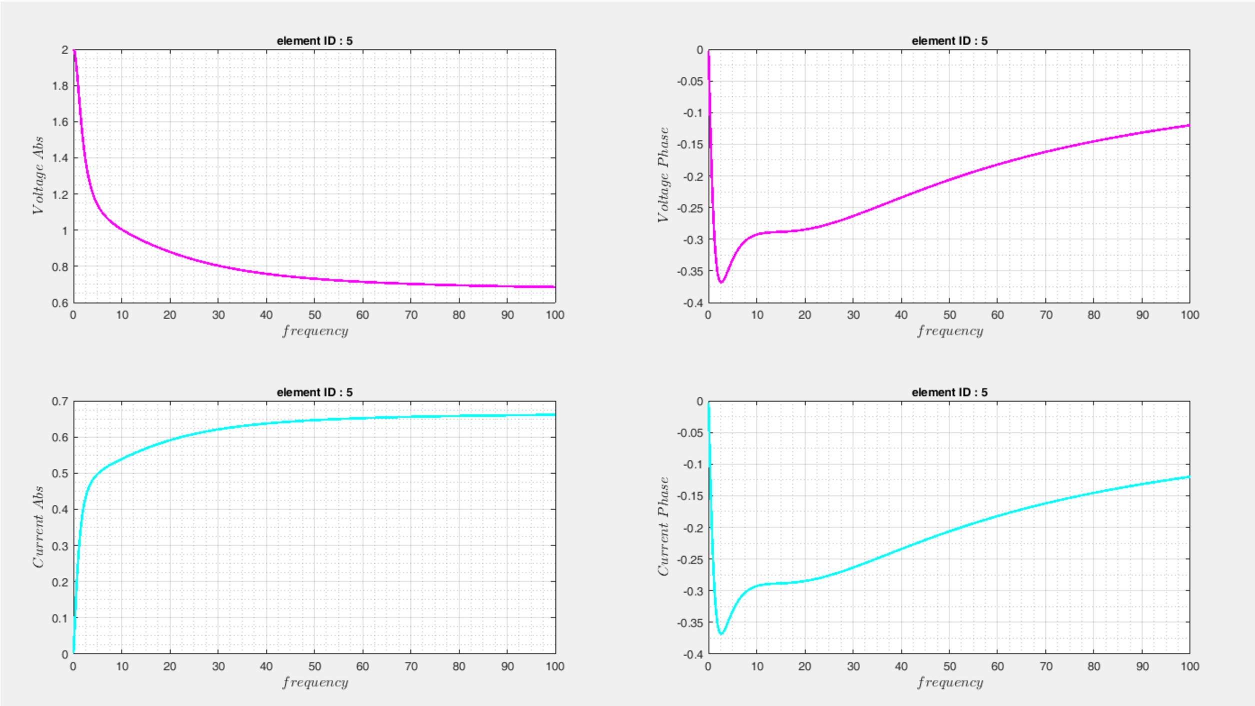

element ID : 4
Voltage Abs
frequency
element ID : 4
Voltage Phase
frequency
element ID : 4
Current Abs
frequency
element ID : 4
Current Phase
frequency

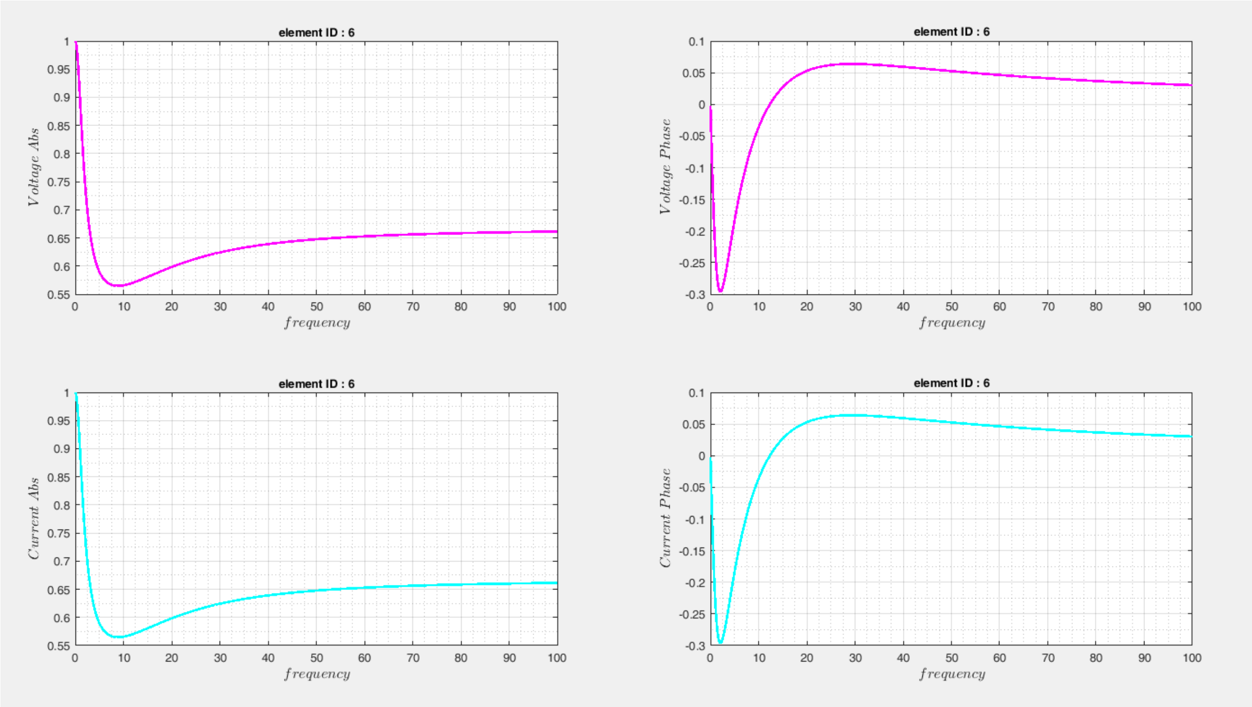


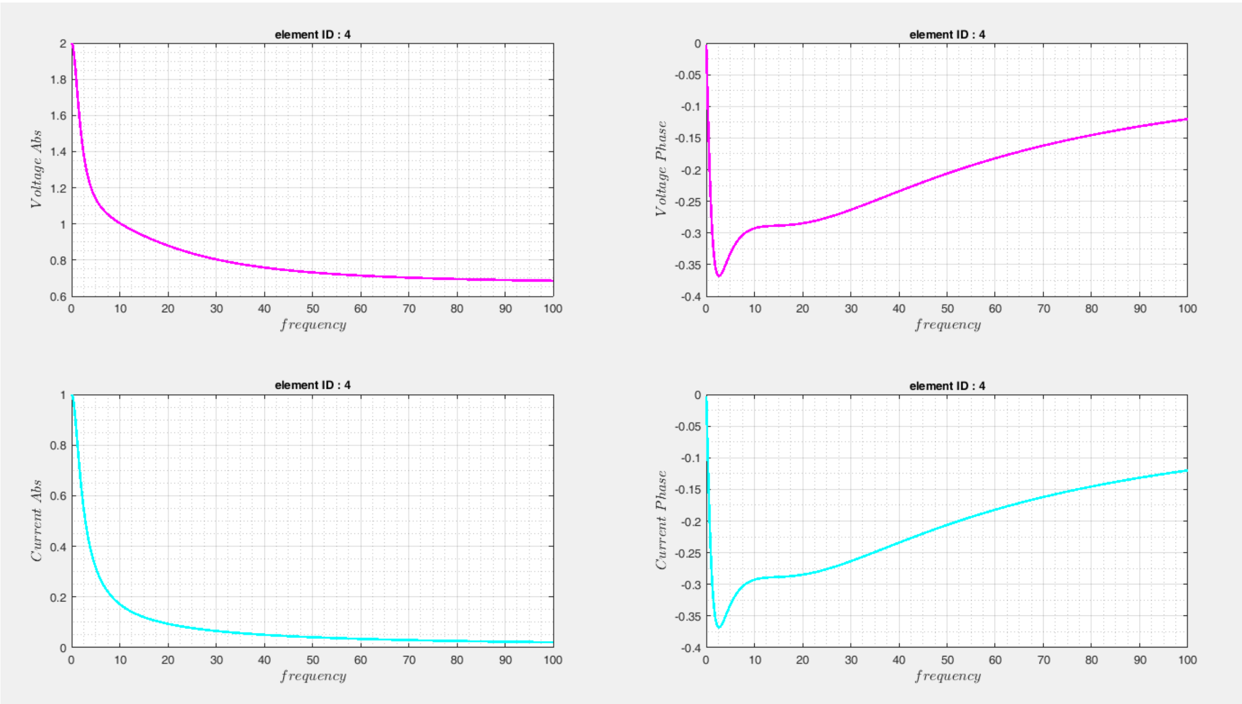
Problematic Source

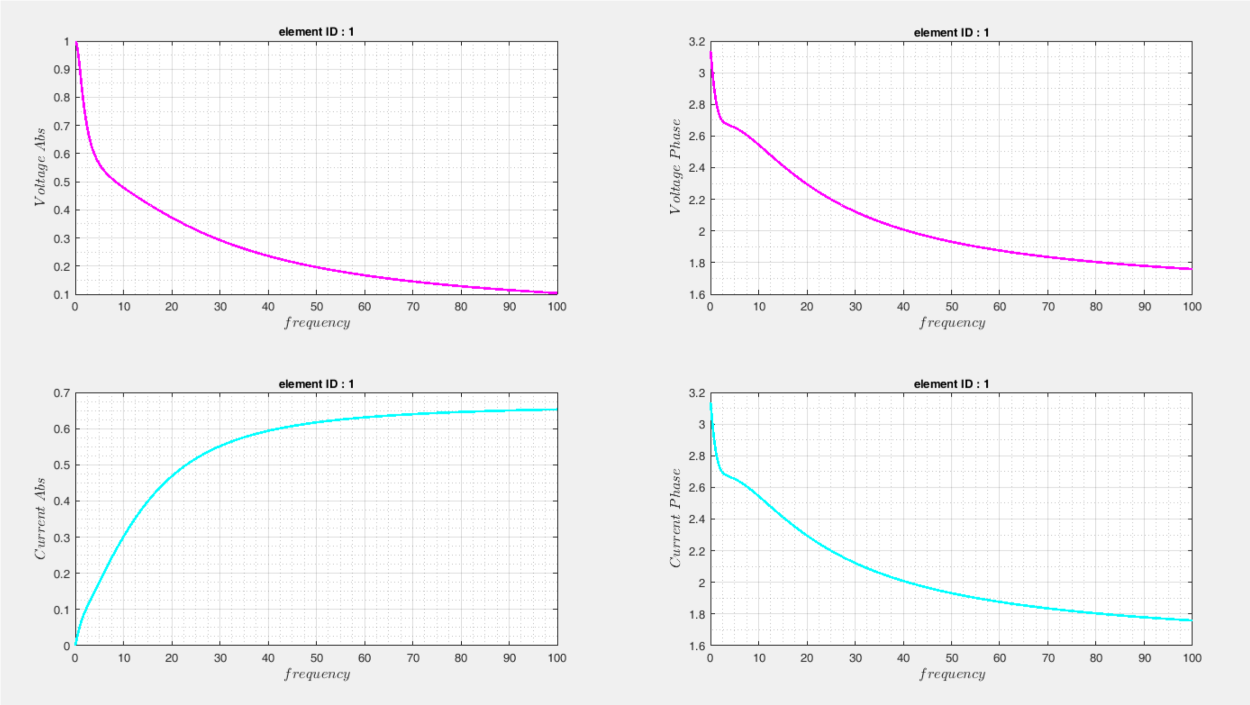


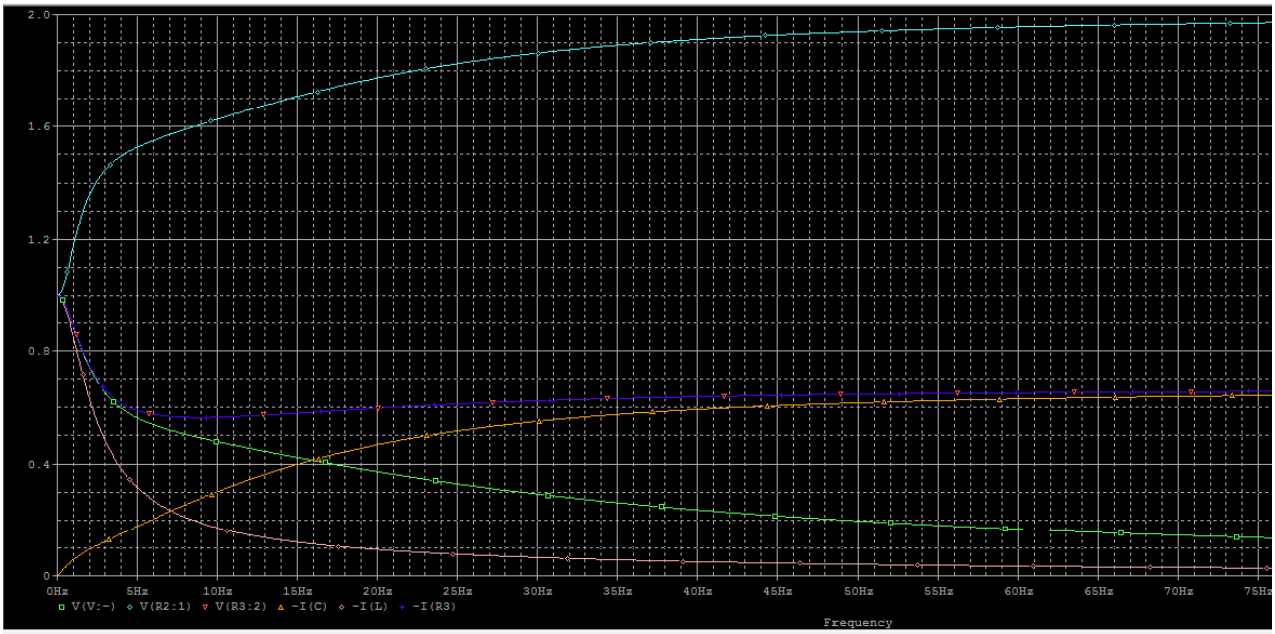

element ID : 2
Voltage Abs
Voltage Phase
Current Abs
Current Phase
frequency


element ID : 5
Voltage Abs
Voltage Phase
Current Abs
Current Phase
frequency

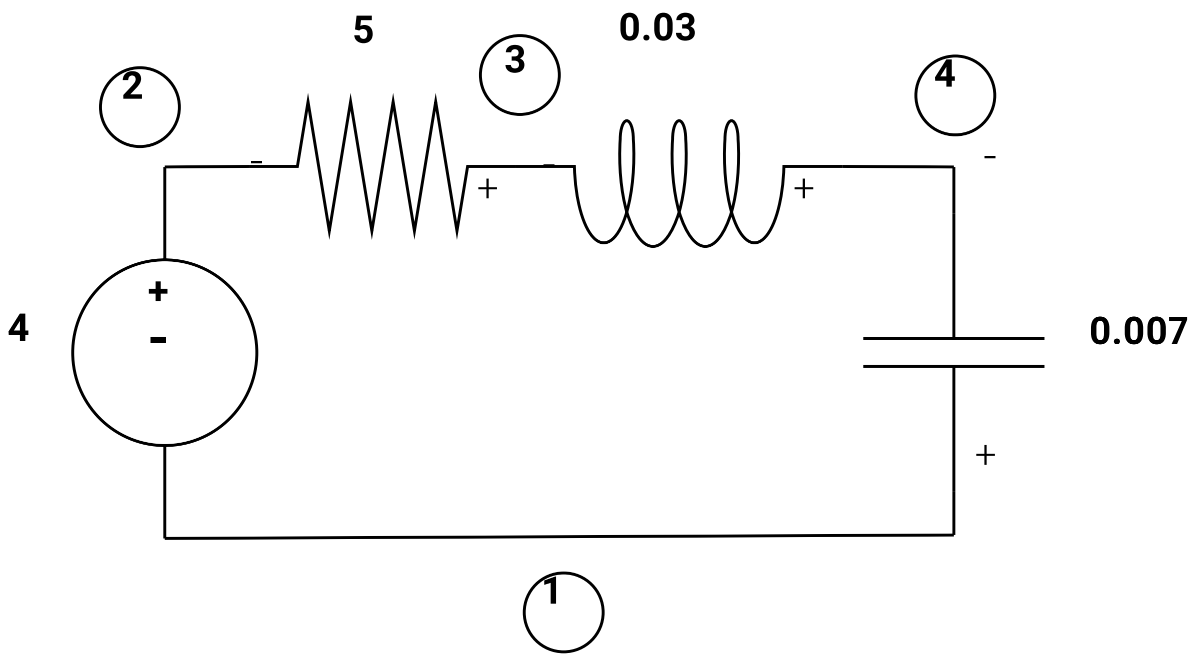

element ID : 6
Voltage Abs
frequency
element ID : 6
Voltage Phase
frequency
element ID : 6
Current Abs
frequency
element ID : 6
Current Phase
frequency

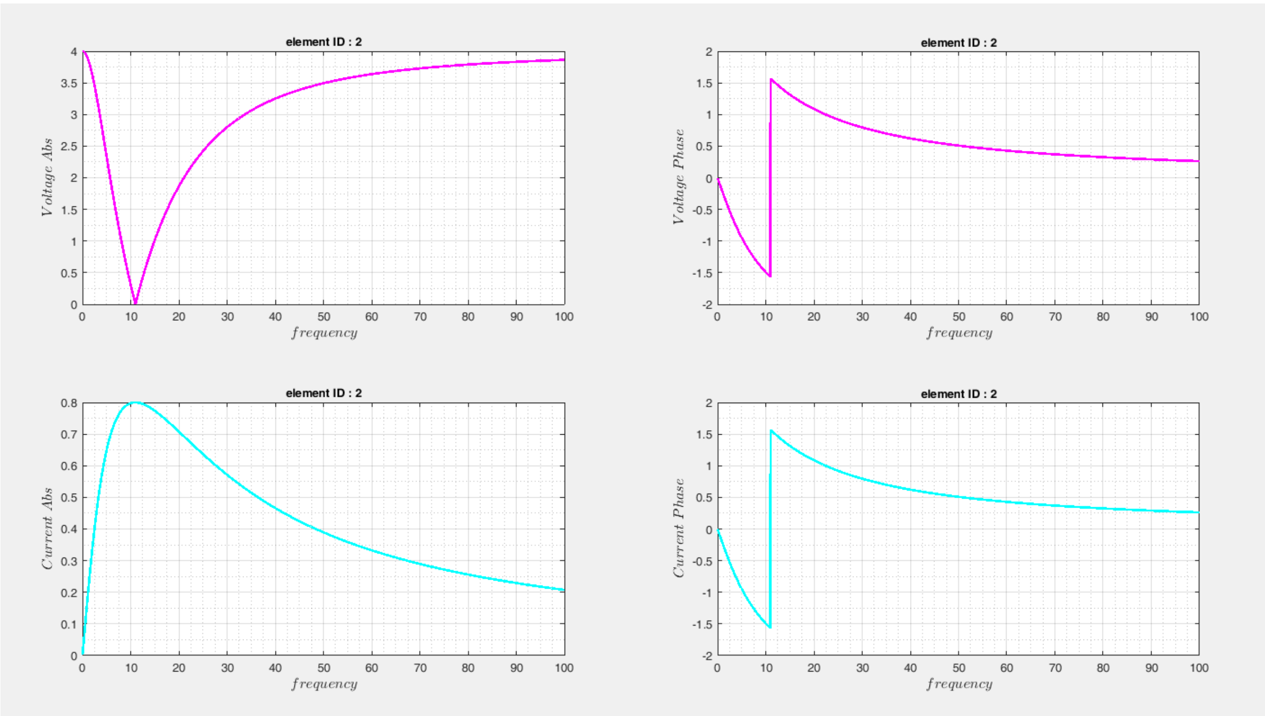

element ID : 4
Voltage Abs
frequency
element ID : 4
Voltage Phase
frequency
element ID : 4
Current Abs
frequency
element ID : 4
Current Phase
frequency

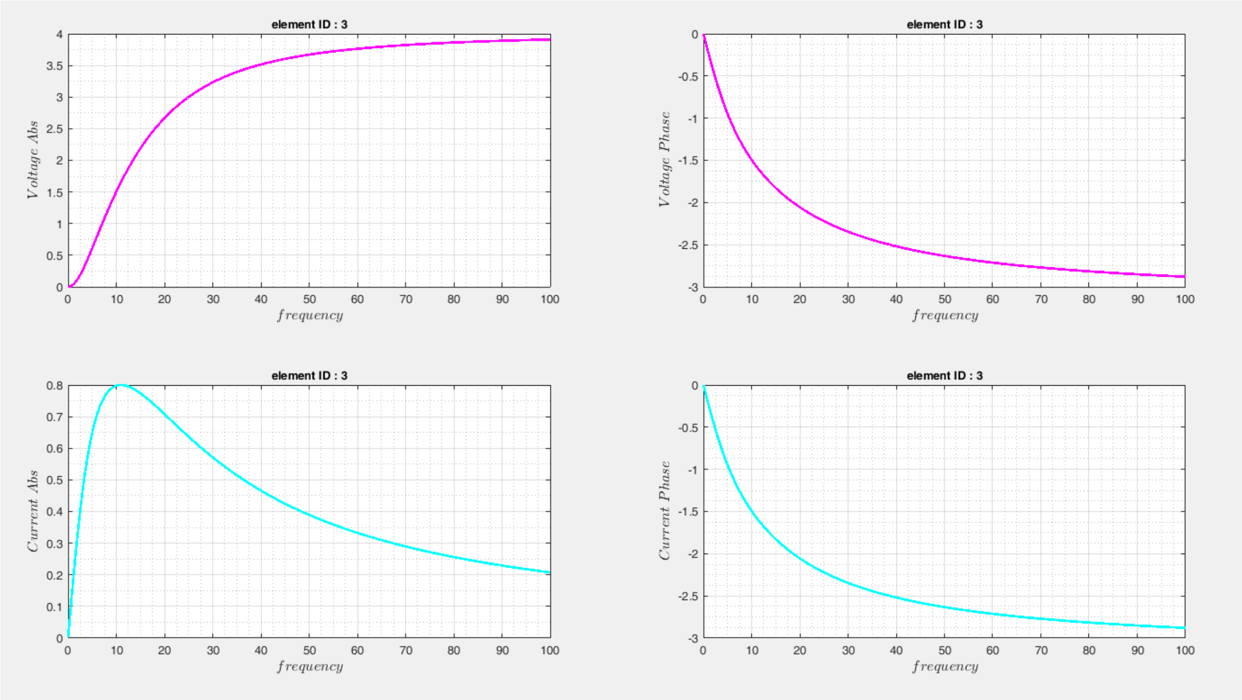

element ID : 1
Voltage Abs
frequency
element ID : 1
Voltage Phase
frequency
element ID : 1
Current Abs
frequency
element ID : 1
Current Phase
frequency

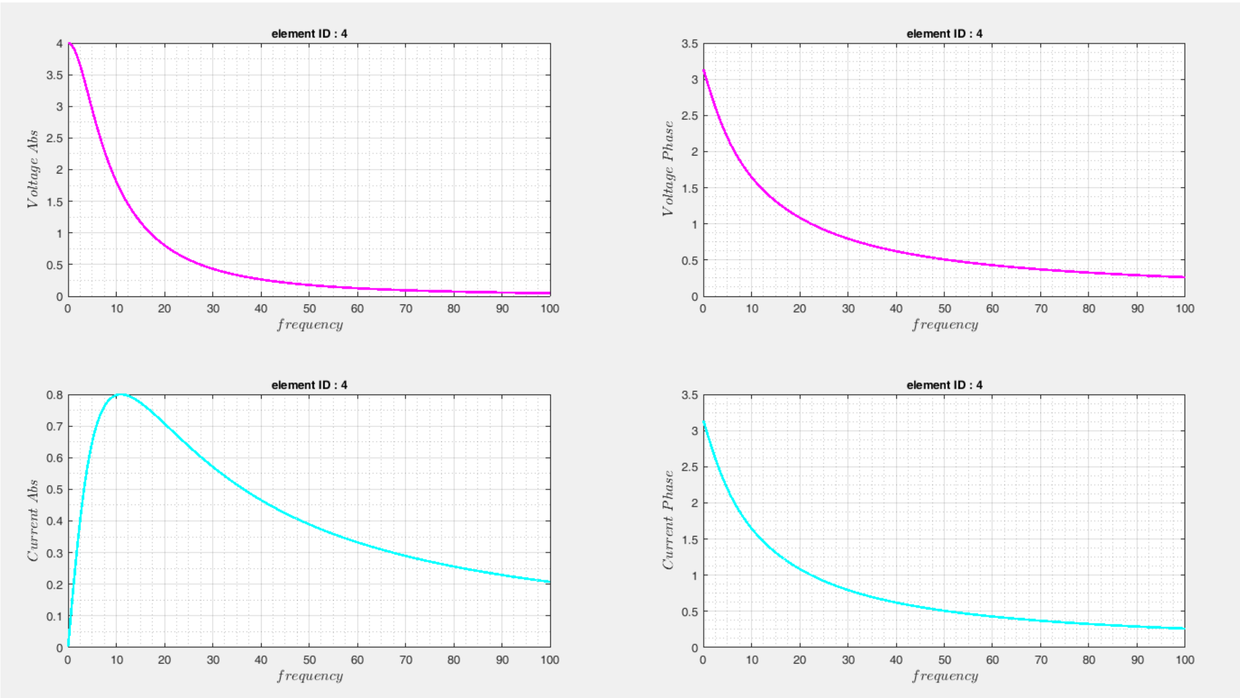

0Hz 5Hz 10Hz 15Hz 20Hz 25Hz 30Hz 35Hz 40Hz 45Hz 50Hz 55Hz 60Hz 65Hz 70Hz 75Hz
V(V:-) V(R2:1) V(R3:2) -I(C) -I(L) -I(R3)
Frequency

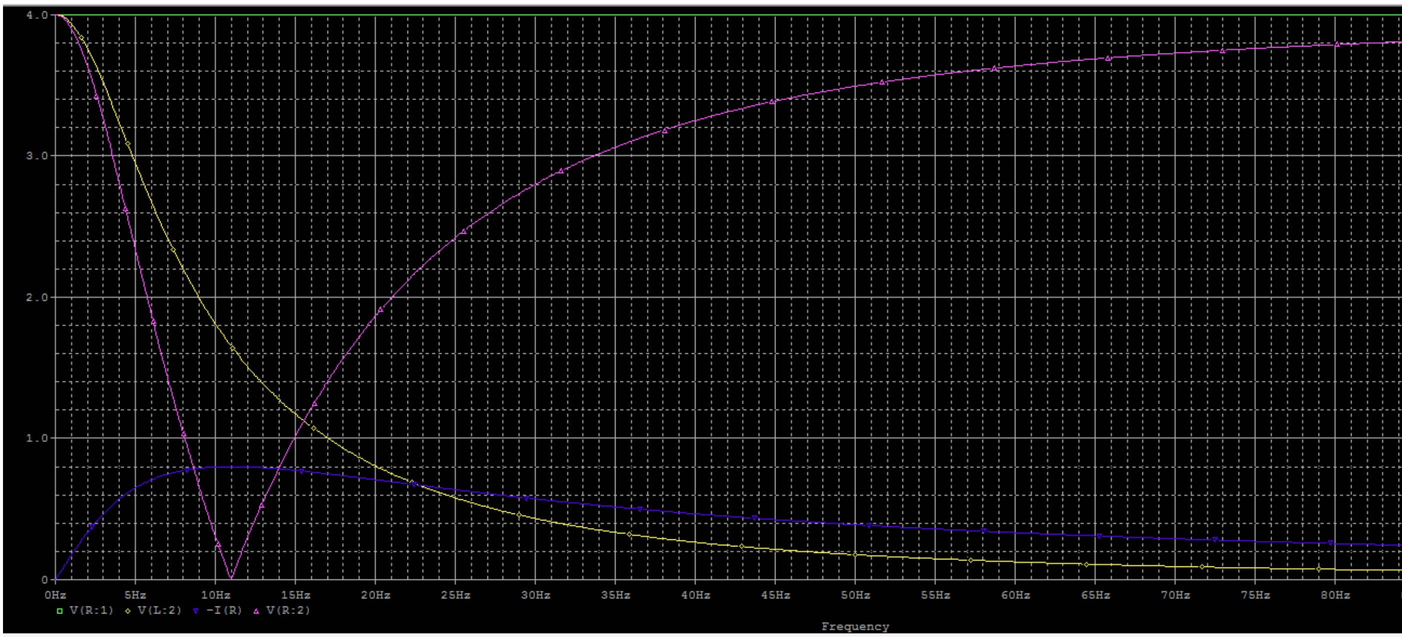
RLC Series


5
0.03
2
3
4
4
0.007
1

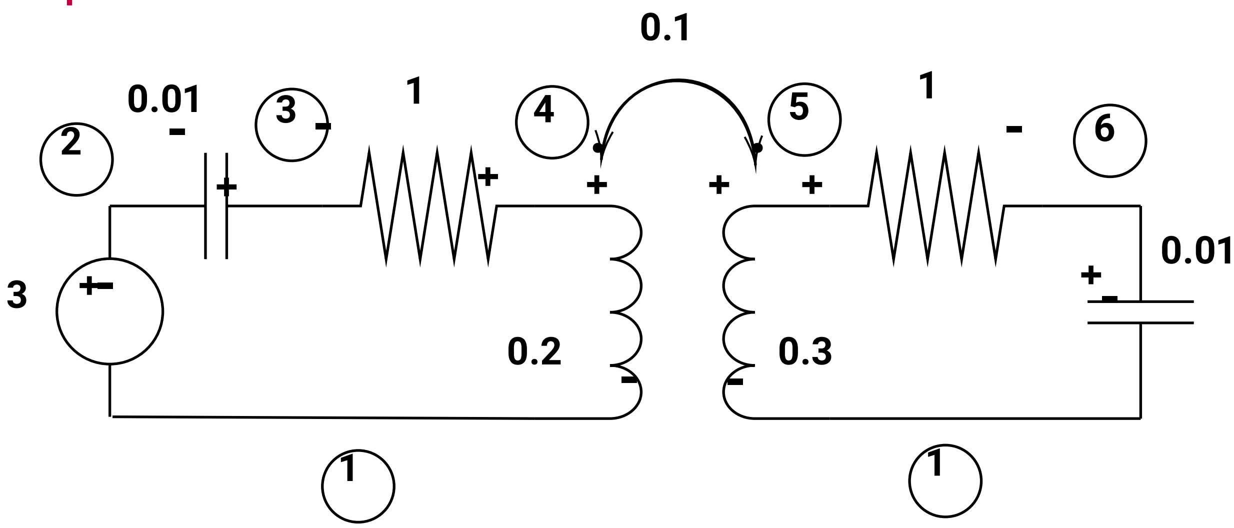

element ID : 2
Voltage Abs
Voltage Phase
Current Abs
Current Phase
frequency

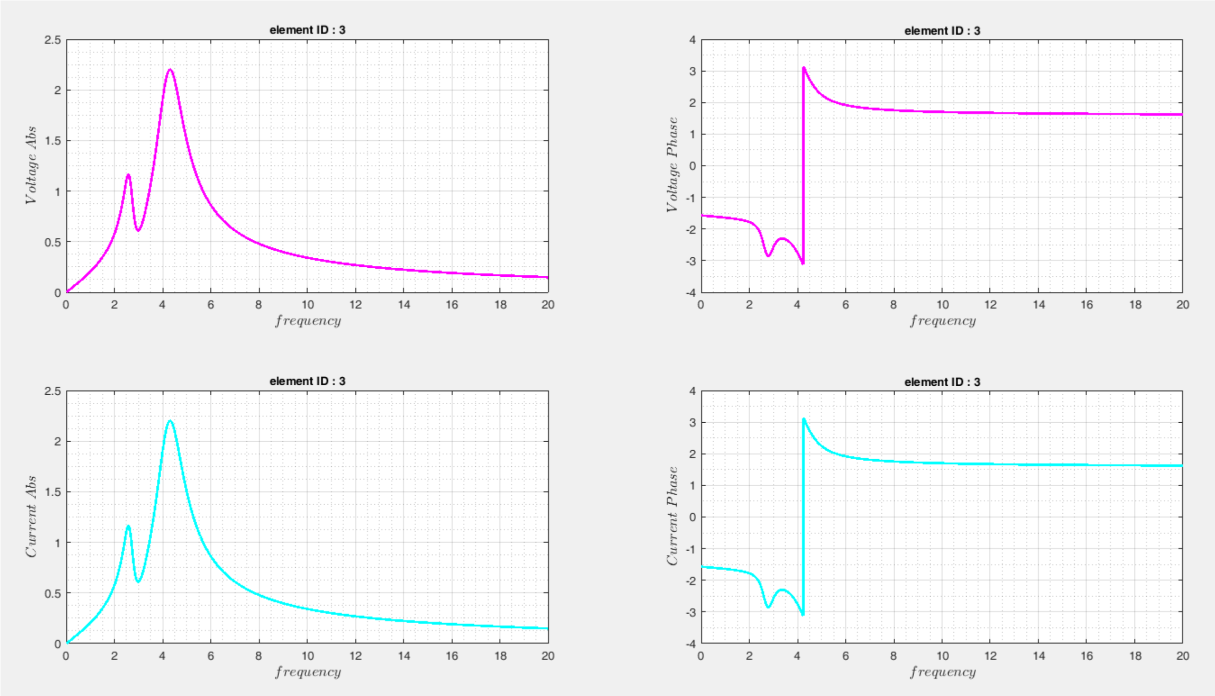

element ID : 3
Voltage Abs
frequency
element ID : 3
Voltage Phase
frequency
element ID : 3
Current Abs
frequency
element ID : 3
Current Phase
frequency

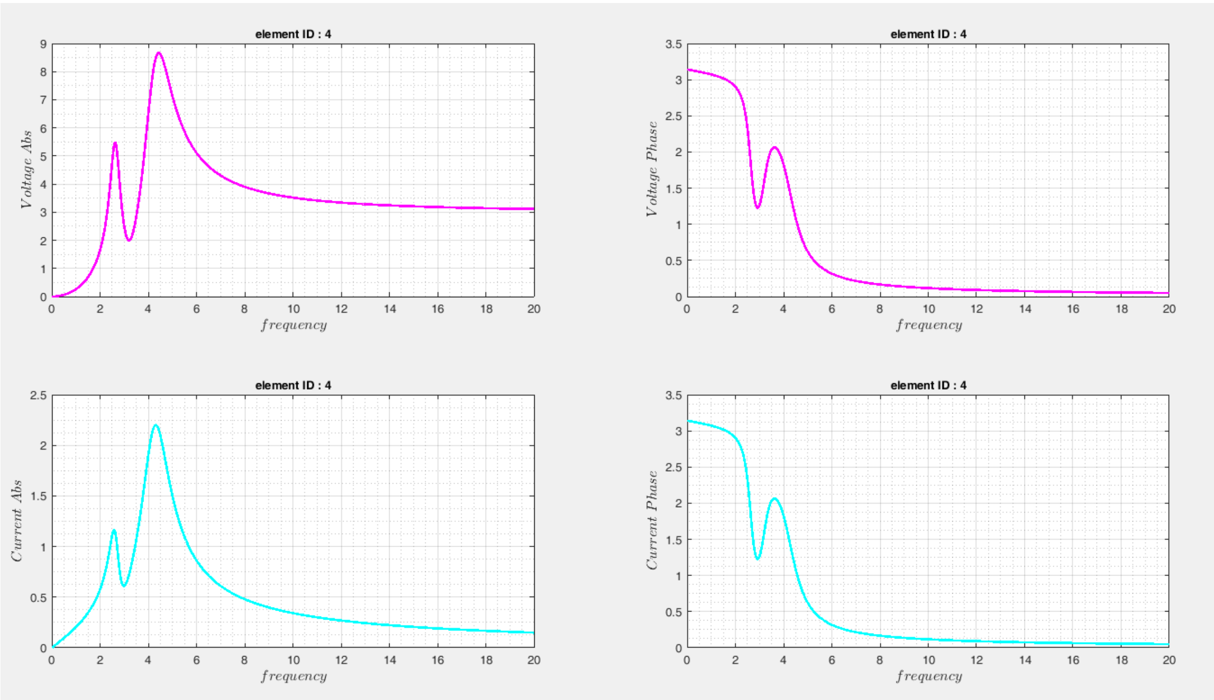

element ID : 4
Voltage Abs
frequency
element ID : 4
Voltage Phase
frequency
element ID : 4
Current Abs
frequency
element ID : 4
Current Phase
frequency

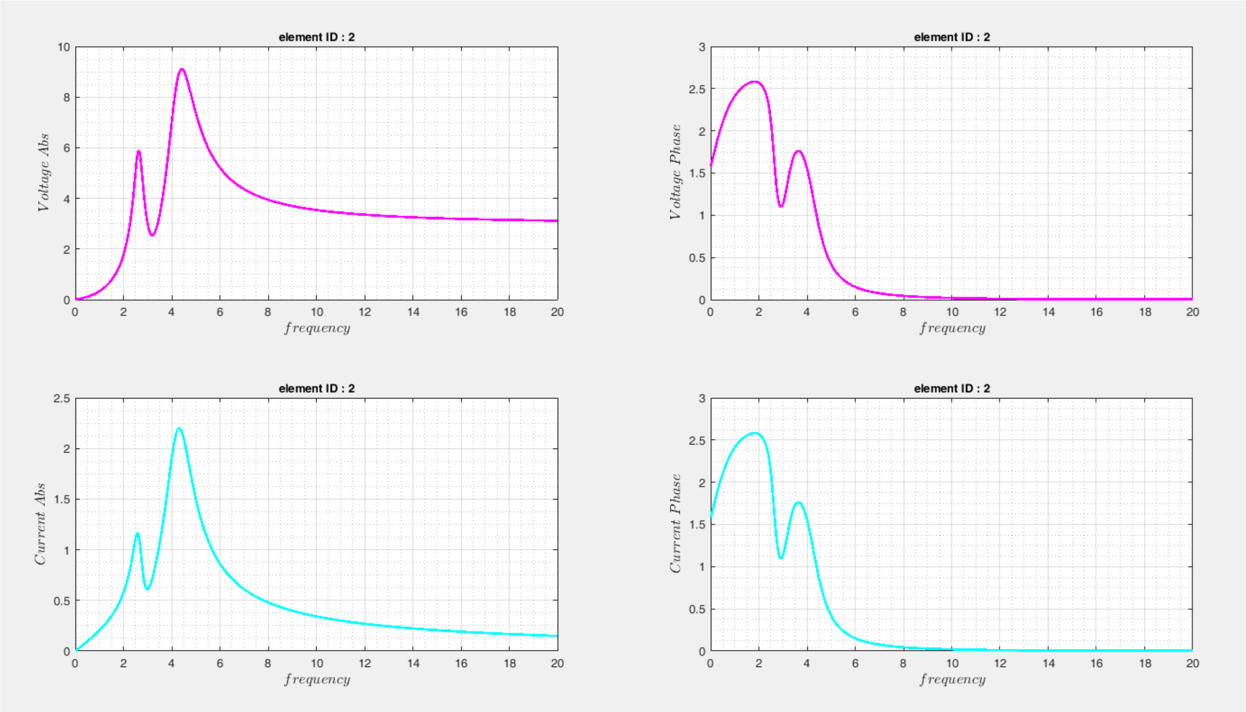


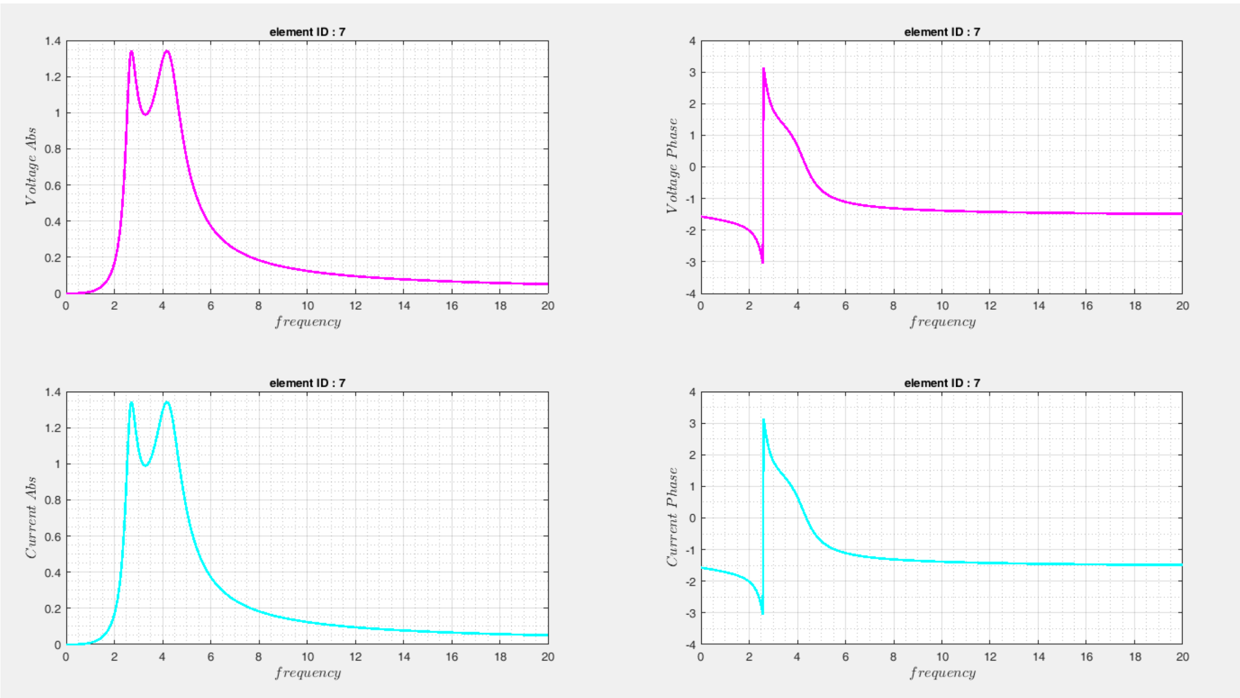
Coupled Circuit

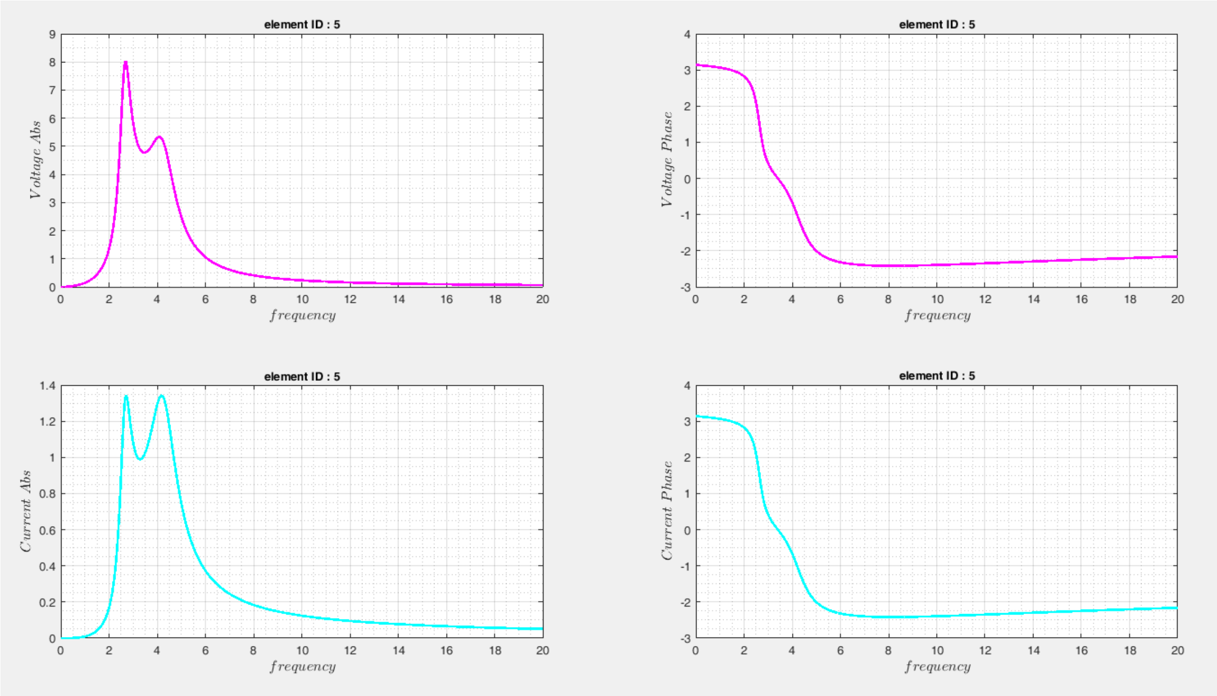


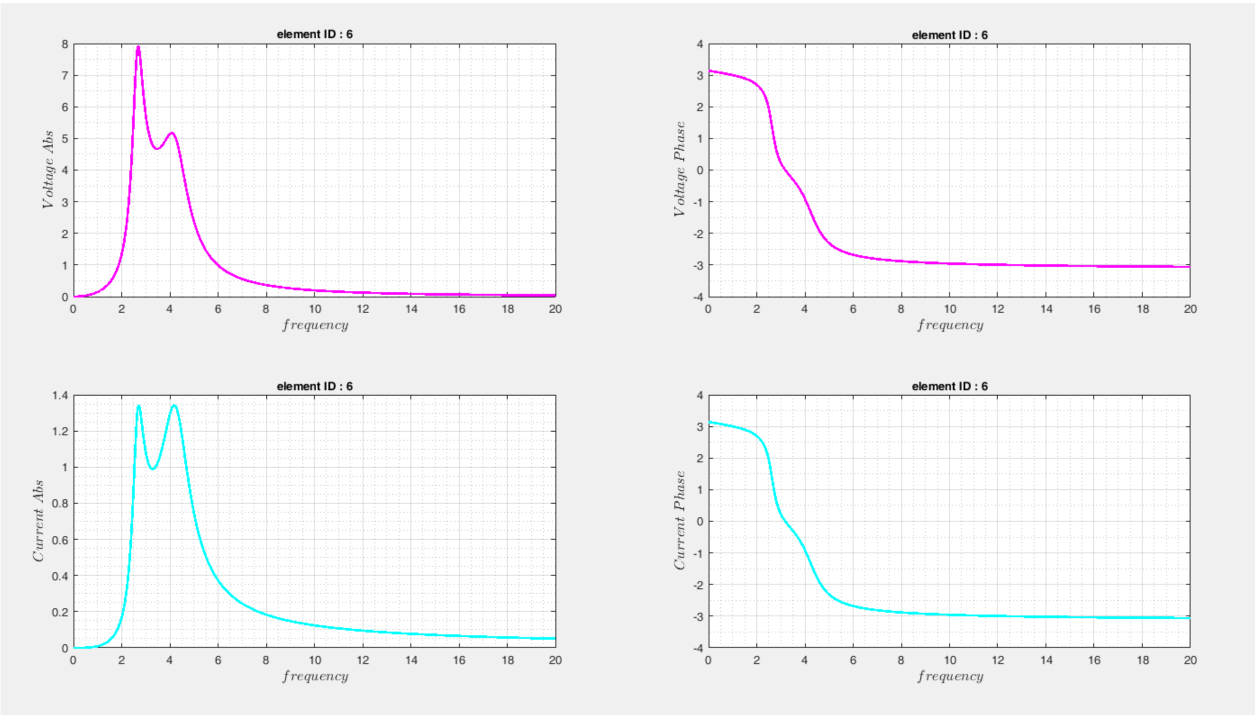

element ID : 3
Voltage Abs
frequency
element ID : 3
Voltage Phase
frequency
element ID : 3
Current Abs
frequency
element ID : 3
Current Phase
frequency


element ID : 4
Voltage Abs
frequency
element ID : 4
Voltage Phase
frequency
element ID : 4
Current Abs
frequency
element ID : 4
Current Phase
frequency

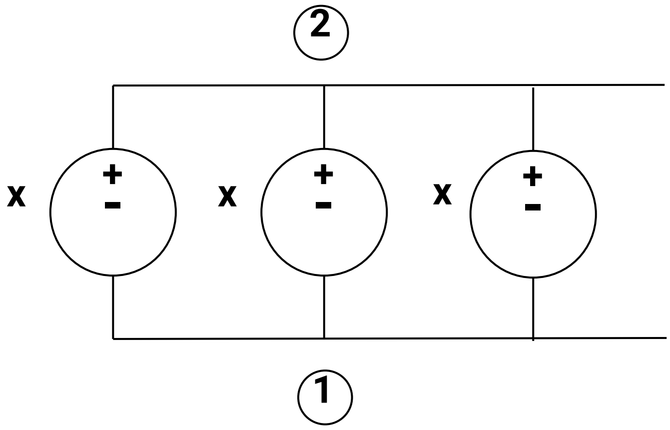

element ID : 2
Voltage Abs
frequency
element ID : 2
Voltage Phase
frequency
element ID : 2
Current Abs
frequency
element ID : 2
Current Phase
frequency


element ID : 7
Voltage Abs
frequency
element ID : 7
Voltage Phase
frequency
element ID : 7
Current Abs
frequency
element ID : 7
Current Phase
frequency

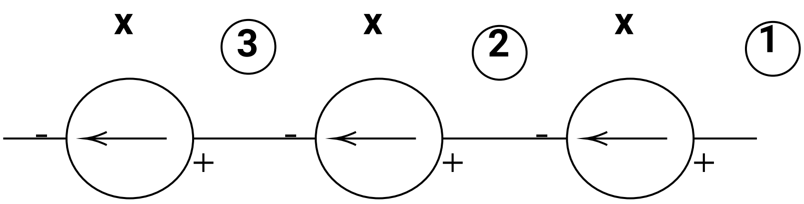




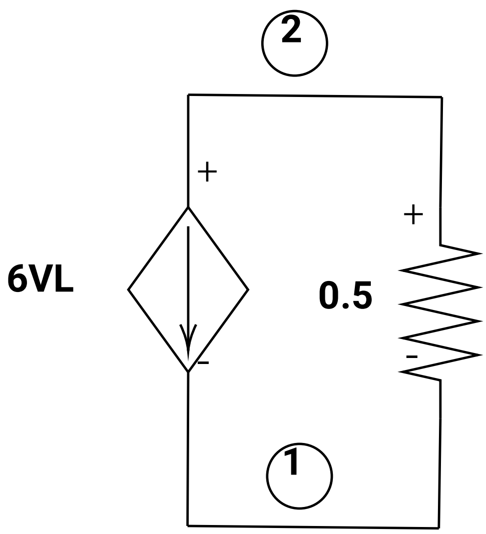
Parallel Voltage Sources

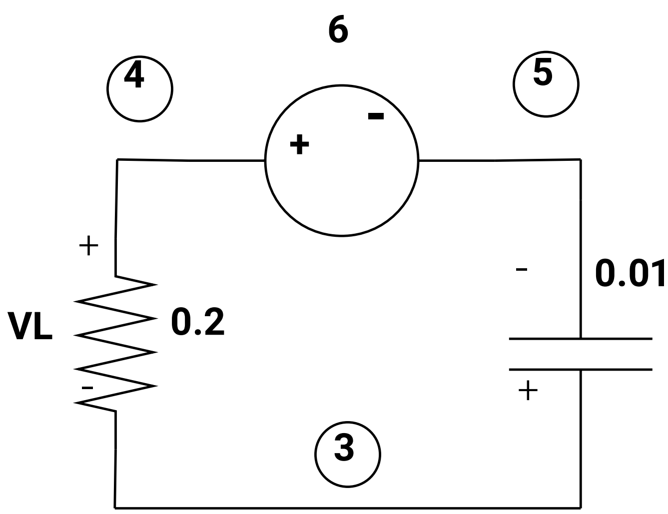


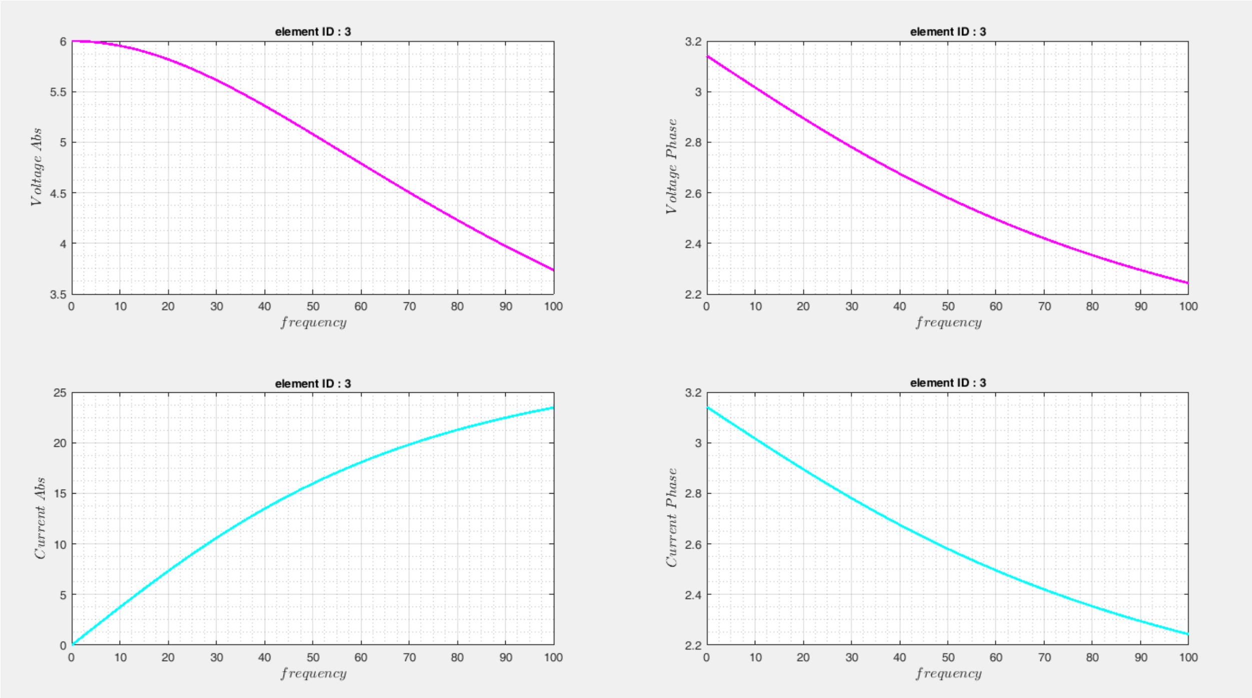
Series Current Sources

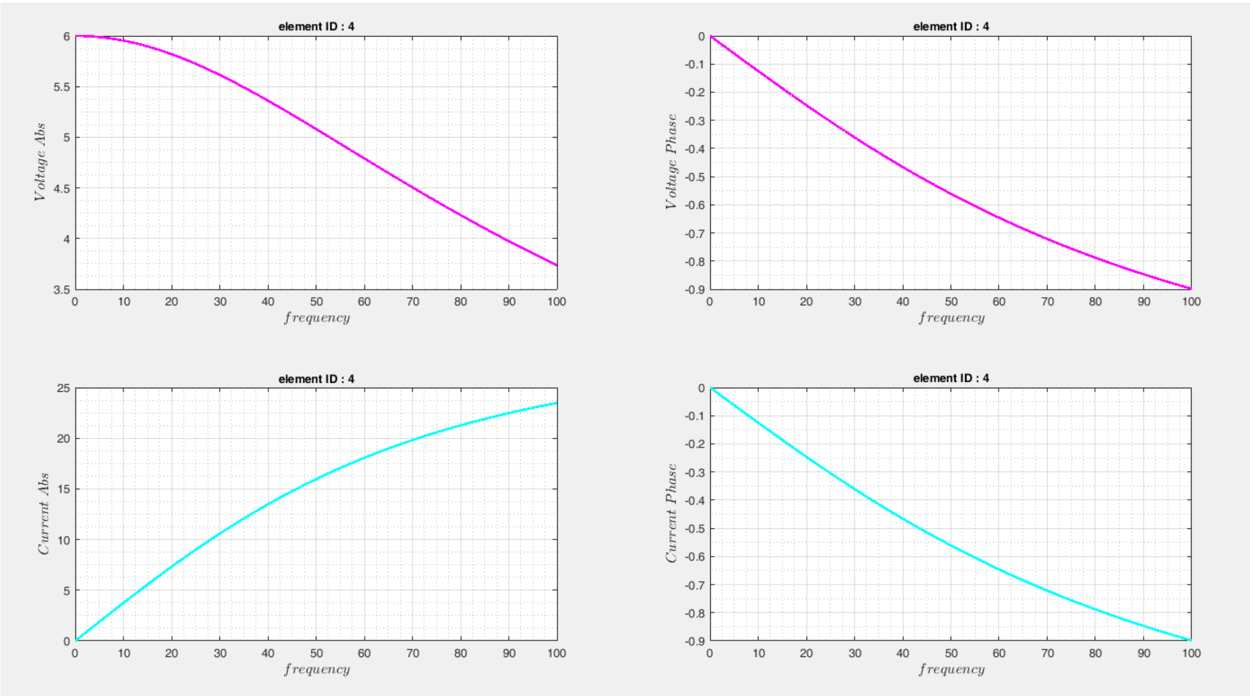


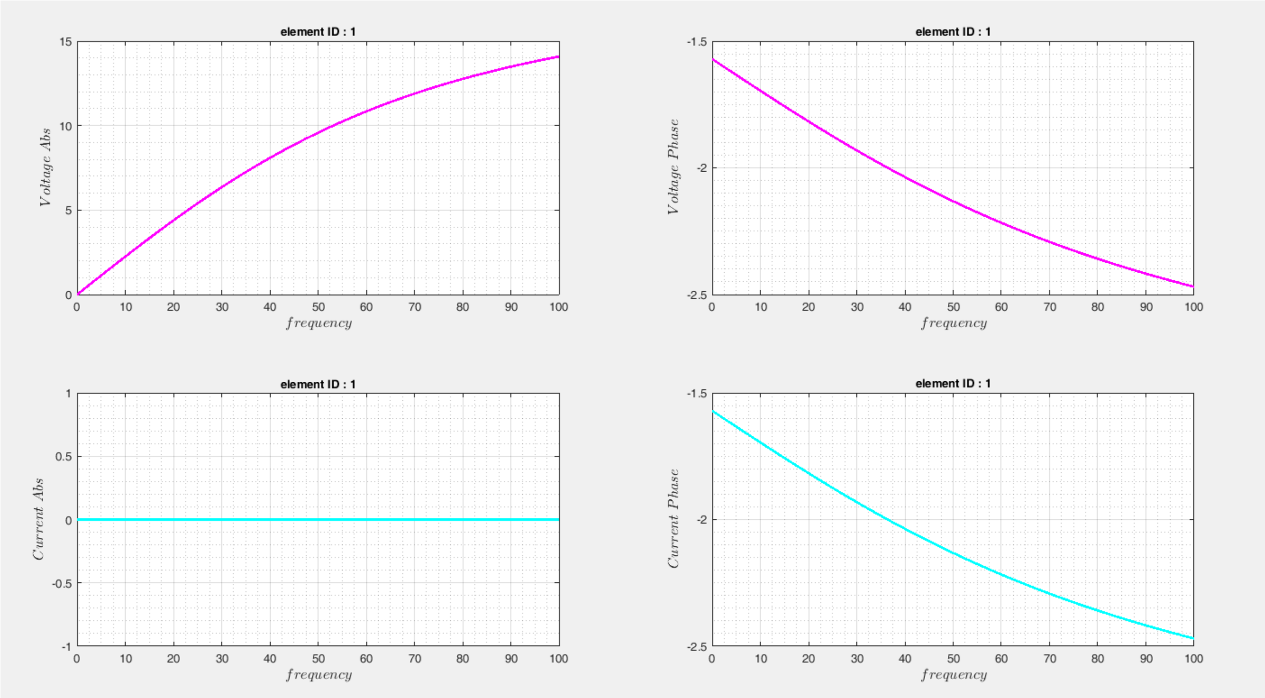
Dependent Source

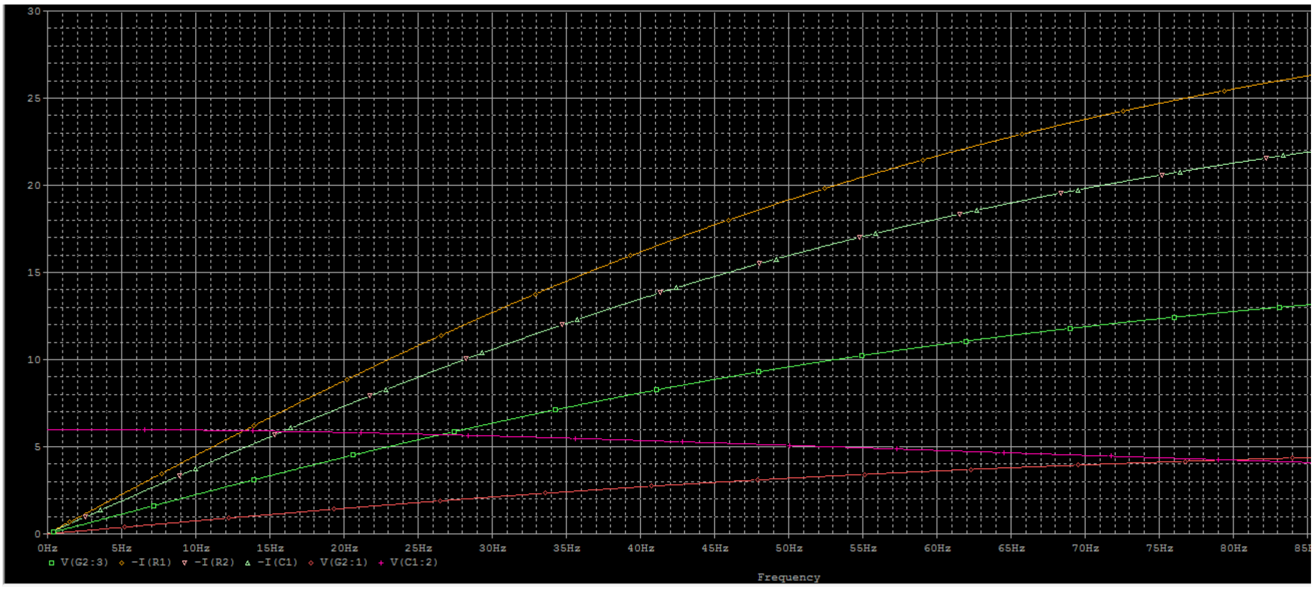





element ID : 3
Voltage Abs
frequency
element ID : 3
Voltage Phase
frequency
element ID : 3
Current Abs
frequency
element ID : 3
Current Phase
frequency


element ID : 4
Voltage Abs
frequency
element ID : 4
Voltage Phase
frequency
element ID : 4
Current Abs
frequency
element ID : 4
Current Phase
frequency


element ID : 1
Voltage Abs
frequency
element ID : 1
Voltage Phase
frequency
element ID : 1
Current Abs
frequency
element ID : 1
Current Phase
frequency


V(G2:3) -I(R1) -I(R2) -I(C1) V(G2:1) V(C1:2)
Frequency