

SPICE SIMULATOR REPORT:

Introduction:

SPICE is a circuit simulator. There are many different variants of SPICE that are used in Electrical and Electronics, and there are many commercial variants as well. SPICE was primarily designed for Integrated Circuit (IC) design, so there are many features there such as transistor-level simulation and non-linear elements that may not be relevant in the early stages of EE, but at the most basic level, it needs to be able to solve Kirchhoff's laws to find the voltages and currents in a circuit.

Code Implementation:

- i) first we are importing modules of numpy as np and os
 - ii) we are creating a function called evalSpice() which Reads the SPICE file, parses the components, and solves the circuit using MNA.
 - iii) i had created a empty list called which is known as parts to store the data
 - iv) The function first checks if the file exists. If not, it raises a "File not found error" using filename.
 - v) Then, it reads the content of the file line by line and starts processing the circuit block between `.circuit` and `.end`.
 - vi) after that it iterates in the p_list in the file and removes the "dc" term.
 - vi) The function extracts the relevant component information (resistors, current sources, voltage sources) from the circuit section. The `type`, `c` (node connected to positive terminal), `d` (node connected to negative terminal), and `value` (resistance, voltage, or current) are identified for each component.
- python

vii)after that it checks if it only consists of “V”, “I” , “R”. by checking the type of each component and raising the error if it contains “L” , “C” .

viii)The function then calls `create_matrices()` to build the matrices that will represent the circuit. These matrices (**Z** and **Y**) are used to solve the system of linear equations based on Kirchhoff’s laws.

ix)Once the matrices are created, the function solves the linear system using NumPy's `np.linalg.solve()` to find the node voltages and currents.

x)The node voltages and currents are stored in dictionaries `V_dict` for voltages and `current_dict` for currents. These dictionaries are returned as the final result.

xi)fixing the voltage value of GND as “0” .

xii)`create_matrices(parts)` : This function builds the matrix representations **Z** and **Y** of the circuit based on the list of circuit components given to it.

xiii)first we create empty lists for current sources ,resistors and voltage sources.set for nodes because to not allow duplicate elements

xiv) we add the node into nodes set if it is not “GND”.

xv)The circuit components are classified into resistors (**R**), voltage sources (**V**), and current sources (**I**). The nodes are identified and stored in respective lists.

xvi)then we will convert nodes set into a list and declare matrices of **Z** and **Y** matrices with corresponding sizes.

xvii)The nodes connected by the resistor are identified as **c** and **d** and if the resistor is connected to a node **c**, the diagonal element $Z[c, c]$ is incremented by $1/R$. similarly for **d**.

xviii)If both **c** and **d** are present, the off-diagonal elements $Z[c, d]$ and $Z[d, c]$ are decremented by $1/R$. This represents the conductance between nodes **c** and **d**.

xix)similarly we can find net current through nodes .The current is subtracted or added to the nodes based on its direction

xx)Voltage sources are handled by adding new rows and columns to the **Z** matrix, ensuring the correct voltage constraint between the connected nodes.

xxi)make the value as 1 for 1st node and -1 for 2nd node of voltage sources .

xxii) then we will return the matrix of Z, Y, nodes, V

xxiii) Now we will solve the matrix by `numpy.linalg.solve` and if the matrix is a singular matrix then it returns the `ValueError`.

xxiv) at last it returns voltage and current dictionaries

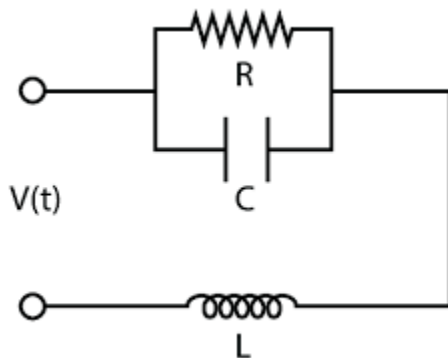
References:

i) i learnt how the `numpy.linalg.solve` function from numpy library works and when to use it in online.

ii) I also referred to how to write an admittance matrix from the internet before writing the code.

Limitation of my code :

i) it won't solve the circuits containing the elements like capacitors ("C") and inductors ("L").



ii) my code won't solve the circuit involving the AC sources it only solve DC sources

$$v = V_m \sin \omega t \quad (4.52)$$

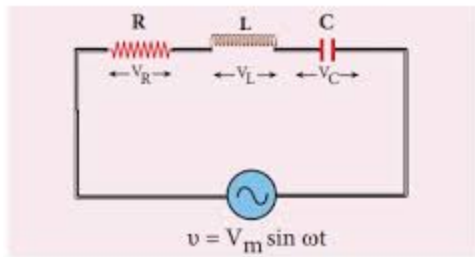


Figure 4.51 AC circuit containing R , L and C

Conclusion :

My code provides an implementation for analyzing simple SPICE circuits containing resistors, voltage sources, and current sources. The solution is obtained using Nodal Analysis with matrix manipulation, and NumPy's `linalg.solve()` is used to solve the system of equations.