SPICE Circuit Solver

V.Manaswini - EE22B065

1 Introduction

The Python script `evalSpice.py` can simulate electrical circuits described in the SPICE format and solve for node voltages and current through branches containing voltage sources using matrix methods. This is essentially a simplified SPICE simulator.

The python script `evalSpice.py` solves for the node voltages and currents through the voltage sources under some circumstances:

- It solves only linear resistive networks
- Circuits with only DC Independent sources
- There should be a reference node, say GND Node
- Doesn't deals with zero resistance resistorss

2 Description

The evalSpice function reads a circuit file and simulates it. It performs the following steps:

1. **Parsing the circuit file**: It reads the file, ignoring the comments and checking for the presence of `.circuit` and `.end` markers. The circuit body is extracted for further processing.
2. **Node Enumeration**: Extracts all the nodes and assigns indices to them. Assuming '`GND`' is marked with index 0.
3. **Matrix Initialization**: Matrices A and B are initialized with all entries as zeros to store KCL and KVL equations of the circuit.
4. **Resistor Equations**: Looping through resistor components, it updates matrix A based on the resistances and nodes they connect.
5. **Current Source Equations**: For current sources, it updates matrix B with the specified current values.
6. **Voltage Source Equations**: For voltage sources, it updates matrix A and matrix B to account for the voltage source values.
7. **Solving**: System of linear equations can be solved by using `np.linalg.solve()`, if there is no solution, it raises an error.
8. **Result Extraction**: Node Voltages and Current through voltage sources are extracted and stored in dictionaries(*nodev* and *nodec*)
9. **Error Handling**: The code handles errors such as missing ground node, invalid components, and unsolvable circuits.

3 Representation of a circuit

Consider an example circuit given below:

```
.circuit
V1 n1 GND dc 10
R1 GND n1 9
.end
```

4 Matrix Formation

The python script setups two matrices $A^{n \times n}$ and $B^{n \times 1}$ with all the elements as zeros using `numpy.zeros((n,n))`, where `n` is the sum of number of nodes and voltage sources. Matrix A consists of co-efficients of equations related to the resistor and voltage source, matrix B consists of external currents and voltages.

Let the number of nodes in a circuit and number of voltage sources are `i` and `j` respectively.

In solution vector, the first `i` values corresponds to the node voltages and the remaining values represent the current passing through the voltage sources. So there are '`i`' KCL equations of the form;

$$\sum \frac{V_1 - V_2}{R_{12}} + \sum I_{12} = ExternalCurrents \tag{1}$$

KVL equations will be of the form;

$$V_1 - V_2 = Voltage \tag{2}$$

$Ax = B$, the system of linear equations formed using (1) and (2), `x` is the solution matrix.

The Right hand side of the equations (1) and (2) corresponds to the matrix B.

Observation

In matrix A, the index(`k`) ranging from 0 to $i - 1$, the diagonal elements are positive.

$$[A_{kk}] = \sum G_{kl}, [A_{mn}] = -\sum G_{mn} \tag{3}$$

G_{mn} represents the conductance value of the resistor connected between the node labelled as `m` and `n`.

The index(`p`) ranging from i to $n - 1$ has elements with values either 1 or -1 depending on the orientation of voltage source.

The entries corresponding to the positive node takes 1.0 and the entries corresponding to the negative node takes -1.0 in matrix A.

In matrix B, the first `i` entries are external currents and the remaining entries are voltages.

The equation $Ax = B$ can be solved by using `x = numpy.linalg.solve()`, while solving, the row index and column index corresponds to the GND node are excluded, since the node voltage at GND node is zero.

If `x` exists, node voltage(0) for the GND gets added to `x`.

- The dictionaries returned by evalSpice.py will be in the form of: ({{'n1': 1.0, 'GND': 0.0}, {'V1': -1.0}})

5 Error Handling

- The file should be with extension `.ckt`
- Ground node must be specified
- The circuits with parallel voltage sources and series current sources are considered to be invalid
- The circuits without `.circuit` and `.end` are treated as Malformed circuits
- Components with first character V, I and R are valid, else it will raise an error
- There shouldn't be a component with same names