

SPICE simulation in Python

This assignment gives details on implementing a basic SPICE simulator for electronic circuits in Python.

Approach

- In the `make_list_of_nodes` function, I started by reading the lines of the input file `filename`, and checking whether `.circuit` is present in the file. Then I store the line of the file in a list.
- Then I store the lines between the markers `.circuit` and `.end` in another list and convert each line from a string to list.
- After obtaining the relevant line, I assign all the nodes a number. `GND` is assigned 0 and `n1` is assigned 1 and so on.
- Now the `mat_forming` creates the admittance matrix by first reading the type of component and then the node values in each line.
 - Assigning the coefficients for resistors is done in the function `R_values`,
 - current sources in lines 115–116. Presence of current sources is reflected in the independent matrix (b), and not in the admittance matrix(A).
 - voltage sources in lines 118–137
- The solution of the operation $Ax=b$ is returned as two dictionaries
 - One has the voltage values stored
 - Other one reports the current passing through each voltage source

Error Handling

- All the error cases mentioned the `test_evalSpice.py` file have been dealt with.
 - The `FileNotFoundError` is raised in lines 13–14.
 - If a blank file is entered, a `ValueError('Malformed circuit file')` is raised at line 10.
 - For now, we are only dealing with purely resistive circuits. Therefore, if any other component is present, a `ValueError` is raised in lines 91–92.
 - Singular matrix is reported via a `ValueError` in lines 143–146.
- Some extra cases are taken care of in the code:
 - Negative resistor values raise a `ValueError('Malformed circuit file')`.
 - If resistor value is equal to zero then it is replaced with $1e-10$ and solved.