SPICE simulation in Python

Approach/Algorithm

Step 1. Parsing

- My program starts with first parsing the file into a list of dictionaries called `data_dic` which contains 3 dictionaries, each for a type of circuit element `V,I,R` if any other element present then it raises an error.
- It is done by first simplifying the text file in the `evalSpice` function itself and then passed onto the `parse` function.
- The simplification is done by appending the lines according to their element type `V, I, R` then passing the lists of specific elements into the parse function.
- Parse function then return a dictionary with `Keys circuit element name` and Value as the list containing `Node1, Node2, Value`.
- There is also a function `get_node` to get all the unique nodes into the `node_list`.
- After this it calculates the total unknowns to make the two matrices to solve. It is of the form `Ax = B` where x is the unknown matrix whose value we need A is the admittance matrix and B is the ind matrix.
- The Algorithm to solve for the unknowns is the same as that of the method mentioned by sir in the instructions file.

Step 2. Matrix Building

- After formation of zeroes matrix the 3 functions `current_matrix, resist_matrix, voltage_matrix` adds the respective values into the 2 matrices named above.
  - `resist_matrix` - This function adds (1/R_value) to their respective nodes in the admittance matrix where the index of where to put the value is done by mapping the index from `data_dic` and `node_list`.
  - `current_matrix` - This function adds the value of current source into the ind_matrix.If `I n1 n2 10` given then the convention is current flows form n1 to n2.
  - `voltage_matrix` - This function adds the of 1 in the admittance matrix for current thru the voltage source and also adds the nodal voltage difference equation.

Step 3. Solving the Linear Equation or Matrix

- After the formation of matrices, they are put in the `gauss` function which outputs the unknown nodal voltages as well as the current thru voltage sources.

- Gauss elimination is referenced from the `Hyper Link` -Gauss Elimination.

All the Errors Handled

- My Program will raise `FileNotFoundError` if the path name or file name given is not found.
- It raises `ValueError("Malformed Circuit Error")` if -
    1. If the text/input file is not formatted with proper `./circuit` and `./end`.
    2. If more than `5 elements` of data in the V and I circuit elements and more than `4 elements` in string containing R including their names.
    3. If the value of the given circuit element is not a `float`.
    4. If the value of R element is `-ve`.
    5. If `repetitive` circuit elements are given in text file.
    6. If the sources are `ac` instead of `dc`.
- Comments can be included after the circuit_element line starting with a #.
- It raises `ValueError("Only V, I, R elements are permitted")` if elements except `V, I, R` are given in text file.
- It raises `ValueError("Circuit error: no solution")` if the circuit is formed but the solution cannot be calculated due to the matrix being singular.
- If `R_Value = 0` is given then instead of adding `1/0 = inf` it adds `1e10` which is equivalent to having `1e-10` resistance which is very low.