SPICE SIMULATOR IN PYTHON- DOCUMENTATION

INTRODUCTION: Spice is an online circuit simulator that takes in a circuit as input and solves it to obtain current and voltage values. The necessary equations for solving a circuit come from Kirchoff's laws.

References: link , link

Solving a circuit has two methodologies: The Nodal and the Mesh Analysis. In this code we employ the nodal analysis to solve the circuit. Nodal analysis is performed by first identifying the nodes in a circuit and then performing KCL at every node. When there is a constant Voltage source can let any amount of current to pass through it, we assume an unknown current variable across each Voltage source and perform the nodal analysis, this is known famously as modified nodal analysis.

PROBLEM STATEMENT:

To solve a circuit containing only V sources, I sources and Resistances. Generate a matrix Ax=B to solve for the unknown node voltages and Voltage source currents.

APPROACH : We attempt to generate a matrix equation to solve all the unknowns in the circuit. The unknowns include the nodal voltages and the currents through each independent voltage source. This matrix is generated by plugging in non-redundant linear equations obtained from the nodal and the modified nodal analysis.

NODAL ANALYSIS: Let's assume that at a given normal node all the currents through resistors are leaving.

sum of all currents leaving= -(currents from constant current sources entering node)

$$\sum(\mathrm{Vi} - \mathrm{Vk})/\mathrm{Rik} = -\sum Iik$$

[*where i is a given node and k belongs to set of all nodes. Rik and Iik are stored in Rmatrix and Imatrix respectivel].*

clearly, coefficient of Vi here =$\sum 1/Rik$

of Vk here= -1/Rik

of Ism here =0

MODIFIED NODAL ANALYSIS: Let's again assume that currents leave the given node and few of the independent voltage sources are connected to it.

sum of all currents leaving= -(currents from constant current sources entering the node)

$$\sum \mathrm{Vi} - \mathrm{Vk}/\mathrm{Rik} + \sum \mathrm{Ism} = -\sum Iik$$

[*where i is the given node and k belongsto set of all nodes.Rik and Iik are stored in Rmatrix and Imatrix respectively . Ism is the unknown current through an Voltage Source*]

Clearly, coefficient of Vi here=$\sum 1/Rik$

of Vk here=-1/Rik

of Ism here =1 (If +ve terminal of Voltage source is connected to the node)

=-1(If -ve terminal of Voltage source is connected to the node)

The coefficients are stored in a matrix: x=[Vn1 Vn2 …….Vnk Is1 Is2 Is3…….Ism]T

EXPLANATION OF THE CODE:

The code is divided into multiple functions whose Mechanism are specified as follows:

def collect_input(f): Takes a file pointer as an argument and reads the relevant data alone (from .circuit to .end) returns a dictionary called input-dict (which stores every element against its nodes and value) and another dictionary compo-map (which is a mapping between node names and node numbers. Default GND is assigned 0). This function calls the store input function.

ERROR!: Raises Value Error if the circuit file is not defined properly

def store_input(input_dict,compo_map,line): Takes in the input_dict and updates the compo_map.

ERROR!: Raises Value error when circuital elements are not legitimately defined

def nodal_analysis(Rmatrix,i,row,Imatrix,compo_map,Vsources):

Takes in the Rmatrix, current node(i), next row in resultant matrix, I matrix, compo_map and the number of Voltage Sources(Vsources) as arguments.It modifies the row by updating it with the value of coefficients from nodal analysis procedure as mentioned above.

def modified_nodal_analysis(Rmatrix,i,row,Imatrix,Vmatrix,compo_map,Vsources,Valone,Imap):

Takes in the Rmatrix, current node(i), next row in resultant matrix, I matrix,V matrix, compo_map and the number of Voltage Sources(Vsources), subset of input dict containing V sources only (Valone) and Imap (a dicitonary that stores unknown current across every independent voltage source against a natural number) as arguments. It modifies the row by updating it with the coefficients from modified nodal analysis mentioned above.

def analysis(Amatrix,Bmatrix,Rmatrix,Imatrix,Vmatrix,compo_map,supernodes,Vsources,Valone,Imap):

Takes in Amatrix, Bmatrix, R matrix, I matrix ,Vmatrix ,compo_map, supernodes(a list of all supernodes in the circuit) , Vsources(number of voltage sources), Valone(subset of input_dict containing V elements), Imap(mapping between unknown current variables and voltage sources) as arguments and analysis every node to redirect to nodal_analysis function is node is a normal node and modified_nodal_analysis function if supernode.

def populate(input_dict,compo_map,Vsources):

Takes in input_dict, compo_map and Vsources as argument and populates the matrix by calling the above mentioned function. Finally returns matrices A and B where Ax=B. This function also populates Rmatrix, Imatrix and Vmatrix which are all n X n matrices where n is number of nodes including GND. The ij-th element of any matrix of R,I,Vmatrix is the value of the corresponding circuit element between the i-th and the j-th node. To note: Iij= -Iji , Vij= -Vji , Rij=Rji. If that particular circuit element is not present the value in the corresponding position of matrix is 0.

CORNER CASES HANDLING: If resistances are in parallel, their resultant is put in the Rmatrix. If there is an element that is not forming a complete circuit, the analysis is done assuming open circuit.(0 current through element).No error is thrown.

ERROR!: Raises ValueError if parallel Voltage sources of different values are connected.

def evalSpice(filename): Takes in a filename as input and solves the circuit by calling the aforementioned functions.

ERROR!: Raises ValueError if Series current sources of different magnitudes are connected together(LinAlgError).Raises FileNotFoundError if the file is not found.