Object-Oriented Programming in Circuit Analysis

Xie Zhiqiang & Wang He

@ Electric Circuits

Introduction

Linear algebra shows great power to solve linear system and Matlab is a popular tool in mathematical field especially matrix calculation.

As for steady state of a circuit, we can easily derive a clear relation by Kirchhoff's Laws and Ohm's law from incidence matrix to any other variable we want.

$$A^T \begin{bmatrix} C1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Cn \end{bmatrix} A * \begin{matrix} v1 \\ \vdots \\ vn \end{matrix} = \begin{matrix} i1 \\ \vdots \\ in \end{matrix}$$

However, we'll face trouble for the inconsistency of various components which may spoil our matrix with fine property and hard to solve.

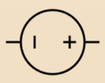
Further more, as for a relatively small system(such as those circuits in our homework), we are always not satisfied with simulating a circuit to check, but to solve the system based on sufficient information.

Functional programming often do lots of trivial work to make itself more robust.

In this project, we tried to develop a framework with a small library based on OOP to solve the system in simple steady state circuit.

Algorithm

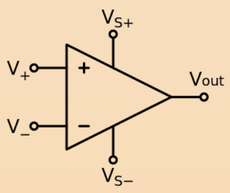
Here is a general way to handle linear component:



As for sources, they just own two properties: V & I and both of them are suited to this matrix.

For the compatibility, we set all explicit current on the right-hand side equation, and any other current implicitly on the solution column so that we can easily call the relation between sources which means dependent sources are covered.

Resistor, capacitor, inductor and wire, these components are concluded in resistance/impedance network. We set all explicit conductance/admittance on the left-hand side matrix and set a current solution in solution column for the same reason as above.



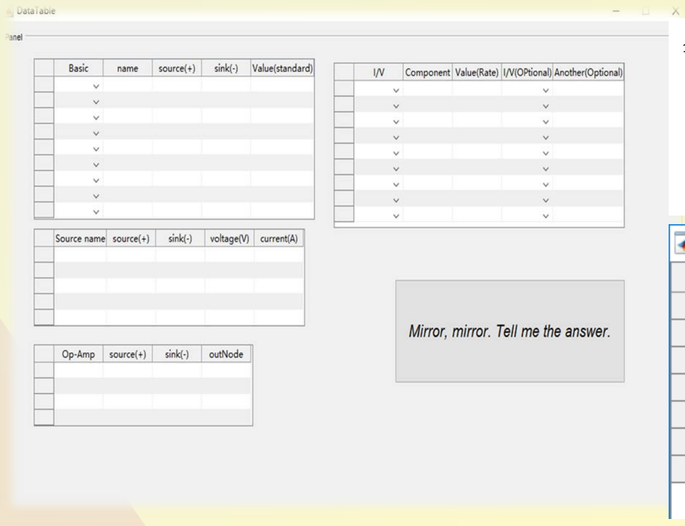
Operational amplifier is special, I just simplify it to 3-pin and ideal, so it's an additional component for circuit. We can express this effect by setting the two input-pins equipotential and set a current solution in solution column.

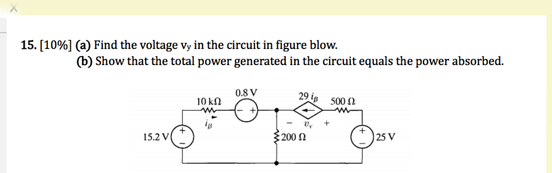
Diode is more important because it's a discrete component as a logical device to control the circuit which may break linear system.
I choose a naïve method to avoid that one node may be locked by two off-state diode: simulate it as a big resistor in off-state and a wire with voltage-low in on-state.
Then iterate all possible state cases and stop in a correct situation.

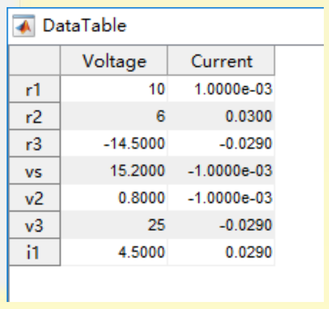
User's manual

Some necessary instructions:

1. Source and sink come from graph theory mean that current flow from source to sink.
2. The value in the first table may represent resistance, capacitance, inductance and threshold voltage of diodes.
3. Node 1 was set as ground, all nodes should be numbered larger than 0;
4. All units are standard.
5. The table on right is for additional information. You can set the relation between different components if you enter the optional column.
6. Yep, input less trivial thing as much as possible, it's not robust yet.
7. It support frequency domain analysis but the w was set as 300 and here is no interface.







DataTable

	Voltage	Current
r1	10	1.0000e-03
r2	6	0.0300
r3	-14.5000	-0.0290
vs	15.2000	-1.0000e-03
v2	0.8000	-1.0000e-03
v3	25	-0.0290
i1	4.5000	0.0290

To be developed

There still exist some improper design even bugs in my program.

1. At the process of preprocess, this program should handle more information to reduce further dynamic allocation though I have cut a lot.
2. The inheritance relationship doesn't play well enough which makes my program bloat of redundant code block.
3. Relaying on cell array is not efficient enough.
4. It's a good tool for simple circuit if I failed this course may I'll use it.

You can get my code from Github

I think it'll be true soon…