Circuit Analysis Based on Object-Oriented Programming and Linear Algebra

Zhiqiang, Xie He, Wang

1 Objective

Write a simple program in Matlab consist of a set of script files, function files and class contents to achieve these goals:

Threat the circuit as a **linear system** so that we can solve it based on enough information.

Components supported:

Resistor, Diode, Wire, Capacitor(phase analysis), Inductor(phase analysis) , DC power supply(current and voltage source, include four controlled sources).

2 Introduction

2.1 Matlab

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language.

In this project we'll put a crucial progress(solve Ax = b) to Matlab to do for its core ability to handle matrix so that we can focus on the circuit itself.

Figure 1: MATLAB

2.2 Circuit Information in Mathamatical perspective

Basic We have learned this inspection method to do node analysis to circuit consisted of current sources and resistor: In general, we can write Gv

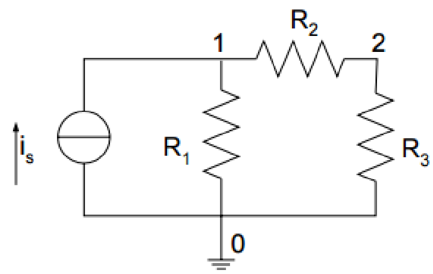


Figure 2: A simple circuit

= i, where G is conductance matrix, v is the node voltage vector and i is the vector of current sources:

$$\begin{bmatrix} (G_1 + G_2) & -G_2 \\ -G_2 & (G_2 + G_3) \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} i_s \\ 0 \end{bmatrix}$$

Substitution Follow the process above, we can solve v_1, v_2 from G_1, G_2 and i_s. But how about solving G_1 or G_2 or i_s if we know other more information offered by other components like the voltage of current source or the current through resistors?

For instance, we set the voltage across the current source as v_s, and suppose the current through R_1 to be i_{R_1}(unknown). Then we can get this equation:

$$\begin{bmatrix} G_2 & G_2 & -1 \\ -G_2 & G_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ i_{R_1} \end{bmatrix} = \begin{bmatrix} i_s \\ 0 \\ v_s \end{bmatrix}$$

Hence, we can solve the system from another perspective not only for simulating from components but also analyse based on state information.

2.3 Object-Oriented Programming

Definition Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which may contain data, in the form of fields, often known as attributes; and code, in the form of procedures, often known as methods.
A feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self").

Apply to components Functional programming often do lots of trival work to handle the inconsistency of various components which makes the whole program less robust and maintainable.
In this project, we tried to develop a framework to construct these various components with proper attributes: source, sink, current, status, etc. You can refer to my code for detailed inforamtion.

3 Implementation for Components

Here are some additional components:

Capacitor & Inductor These two components are for phase analysis only. Convert capacitance and inductance into impedance(jωL or $\frac{1}{j\omega C}$), where 'j' is the imaginary unit and the 'ω' is angular velocity. And then we can treat them as the same as resistor to build impedance network.

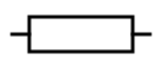


Figure 3: Z_L

Diode Since diode is nonlinear and even discrete in simplification, we set these crucial attributes for diode:

status: on / off indicate whether this diode is 'conductive'.

onResistance: I set forward resistance in 'on' state to a small value($1\mu\Omega$).

offResistance: for simplification, I set both forward and backward resistance to be a large value(100MΩ). To deal with the discontinuity, I iterate all possible states and checking the result until one of them satisfied this:

```
for dio = 1:diodeN
        checkD = diode{dio};
        if checkD.checking()
          if dio == diodeN
            output;
            return;
          end
        else
            break;
        end
end
function check = checking(obj)
        global solutionM
        voltage = solutionM(obj.source) - solutionM(obj.sink);
        if voltage >= obj.threshold
            check = false;
        else
            if obj.status == 0
                check = true;
            else
                check = false;
            end
        end
end
```

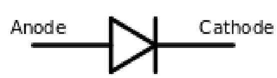
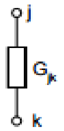


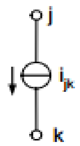
Figure 4: Diode

Conductance connected between nodes j and k



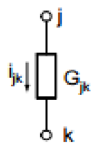
	v_j	v_k	RHS
Node j	G_{jk}	$- G_{jk}$	
Node k	$- G_{jk}$	G_{jk}	

Current source between j and k



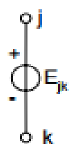
	v_j	v_k	RHS
Node j			$-i_{jk}$
Node k			i_{jk}

Branch current as an output variable



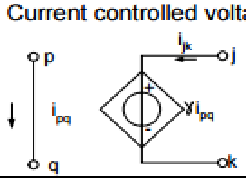
	v_j	v_k	i_{jk}	RHS
Node j			1	
Node k			-1	
Branch jk	G_{jk}	$-G_{jk}$	-1	

Ideal voltage source



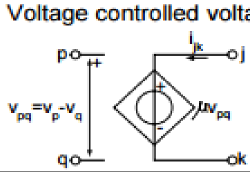
	v_j	v_k	i_{jk}	RHS
Node j			1	
Node k			-1	
Branch jk	1	-1		E_{jk}

Current controlled voltage source



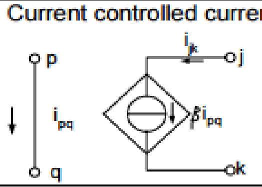
	v_j	v_k	i_{jk}	i_{pq}	RHS
Node j			1		
Node k			-1		
Branch jk	1	-1		γ	

Voltage controlled voltage source



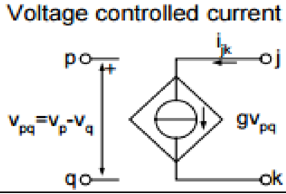
	v_j	v_k	i_{jk}	v_p	v_q	RHS
Node j			1			
Node k			-1			
Branch jk	1	-1		$-\mu$	μ	

Current controlled current source



	v_j	v_k	i_{pq}	RHS
Node j			β	
Node k			$-\beta$	

Voltage controlled current source



	v_j	v_k	v_p	v_q	RHS
Node j			g	-g	
Node k			-g	g	

Figure 5: A summary of the stamps of some basic elements

Operational amplifier In this project, I just simplify it into most basic model: virtual short and virtual-off.

This piece of code explain more clearly:

```
function adding(obj,index)
        global circuitM solutionM knownM opAmp;
        [m,n] = size(circuitM);
        circuitM(obj.outPoint,n+1) = 1;
        circuitM(m+1,obj.source) = 1;
        circuitM(m+1,obj.sink) = -1;
        solutionM(n+1,1) = 0;
        knownM(m+1,1) = 0;
        opAmp{index}.solutionIndex = n+1;
end
```

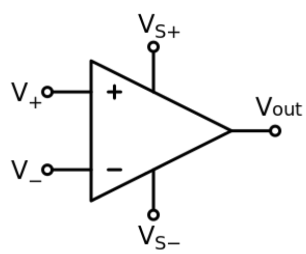


Figure 6: Operational amplifier

4 User's manual

Firstly, run the code in the folder called Circuit analysis. You need to input the command ***DataTable*** to wake up the program.And you will see the UI which contains four tables.

4.1 The first table

In the first table, you need to input the components of the circuit containing resistor, diode, capacitor and inductor. If the circuit is difficult to tell the nodes apart, you can use the wire to connect the components. For each components, our code need some information to know them such as its name, the two nodes which it connects, the value(if it is the diode, the value is its threshold voltage). **Note**:If you click the first column carelessly when no more components needed to be input. You can choose the blank.

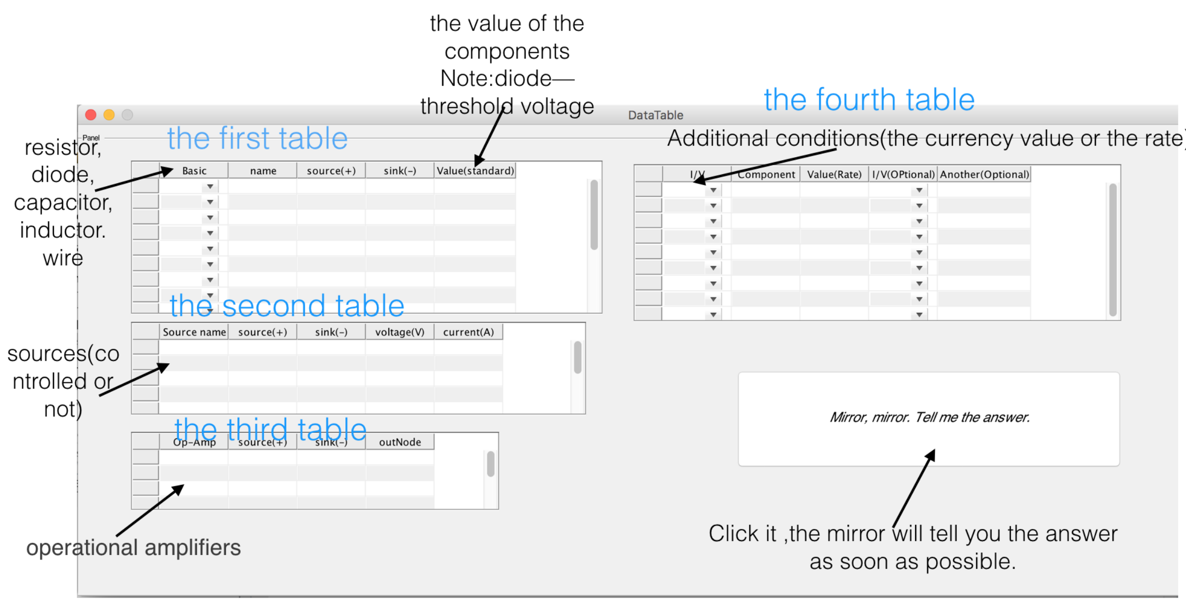


Figure 7: DataTable

4.2 The second table

The second table only includes the sources. Now our code can deal with both controlled source or not.

some tips:

1.current source:the current flow in is the positive electrode.

2.voltage source:the source is the positive electrode and the sink is the negative electrode.

3.If you don't know the accuracy value, just leave it and input other information in the fourth table.

4.3 The third table

In this part you only need to input three nodes (positive electrode, negative electrode, out node) We can deal with the ideal operational amplifiers.

4.4 The fourth table

You can add some additional conditions such as the value or the rate to get the answer or the information which you skipped before.And if you input the voltage or the current of the components, you can ignore the last two columns.However, if you input the rate, you need input more information. For example, if there is a independent voltage controlled current source called i1 which is twice voltage of the resistor r1, the first three columns are about the source (eg:current, i1, 2) , and the last two column are about r1(eg:voltage, r1).

Some necessary instrucions:

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.

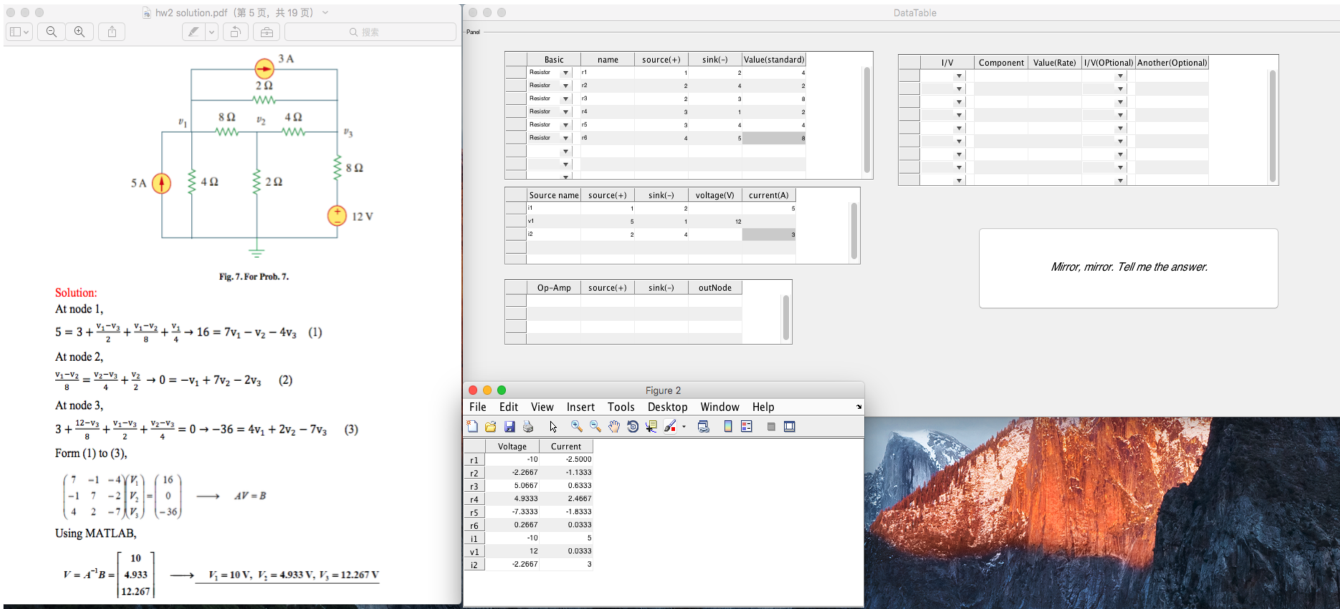


Figure 8: basic test

5 Testing

Here I select most circuit from homework 1-4 to test my program. Which cover most supported components and situation I want to handle. I compare the output result to the homework solutions to prove the correctness of my program. Here are some more explanation:

1. Capacitor and inductor aren't shown here for I just set simple interface for it and I'll explain it more.

2. The precision shown in the table isn't correct because MATLAB modify the output result to adapt GUI but to check the data in cell array.

3. I compare all result to the homework solutions and then analyse my program. All arithmetic accuracy defect come from float point number mathematical operations if here's no diode in circuit. Here you can visit this site for more information about MATLAB float point number: https://cn.mathworks.com/help/matlab/matlab_prog/floatingpointnumbers.html.

4. I'll just show some representative tests here and attach more in appendix.

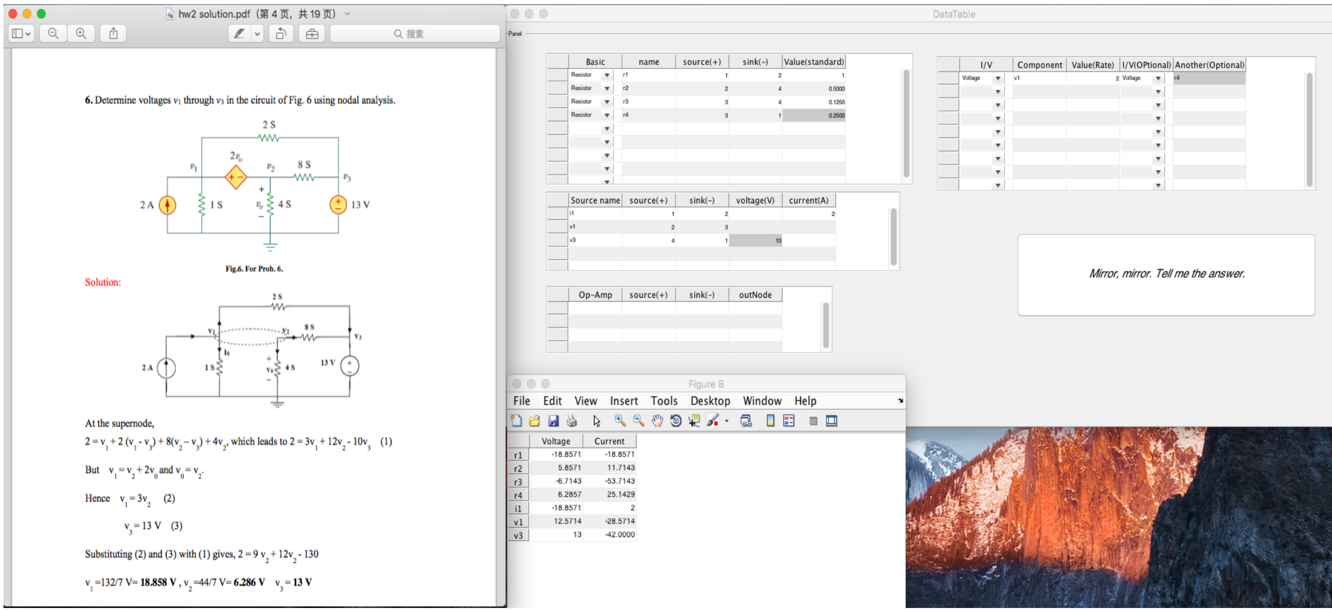


Figure 9: controlled source-0

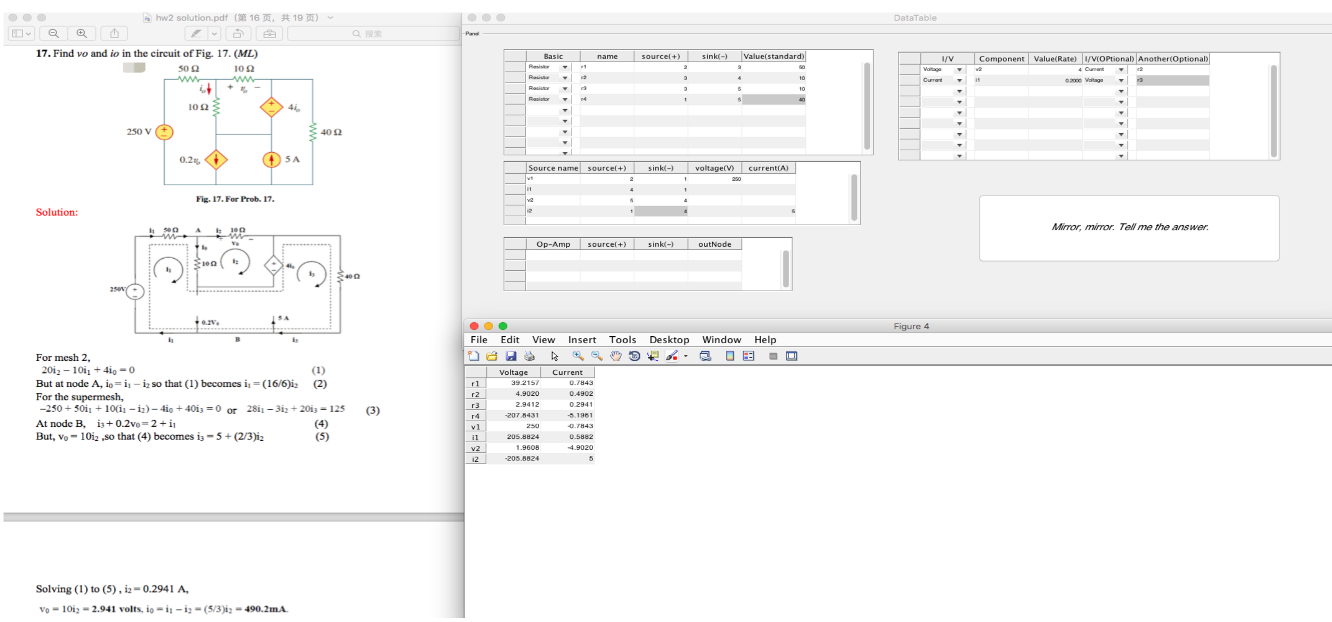


Figure 10: controlled source-1

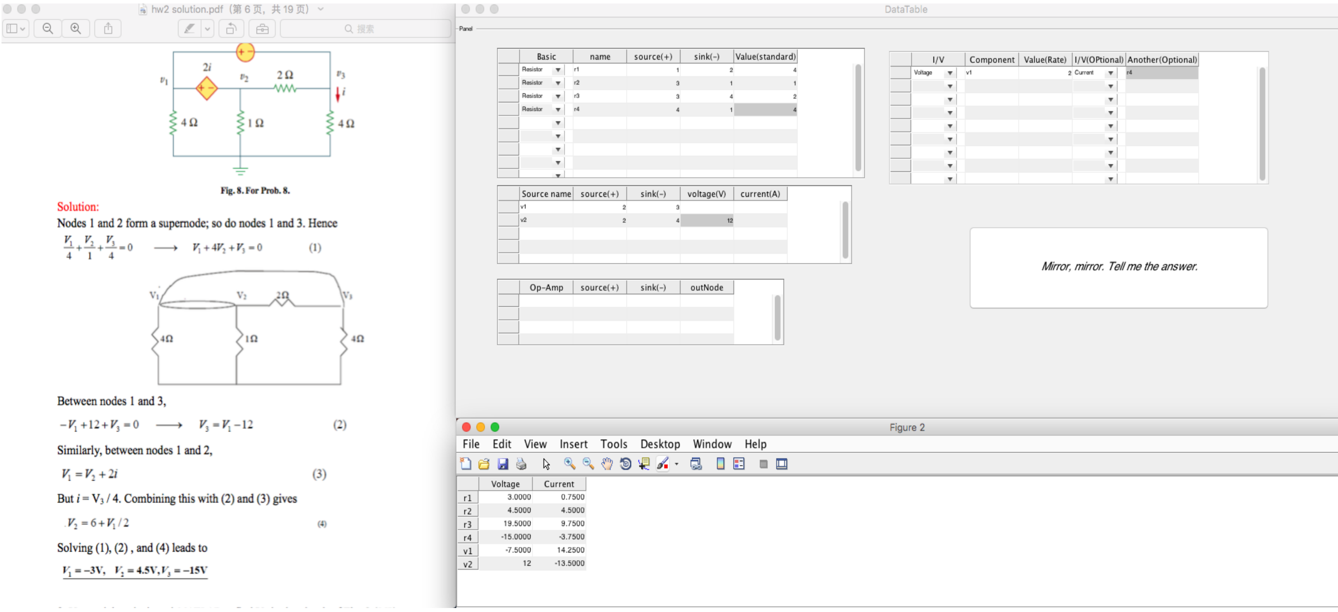


Figure 11: controlled source-2

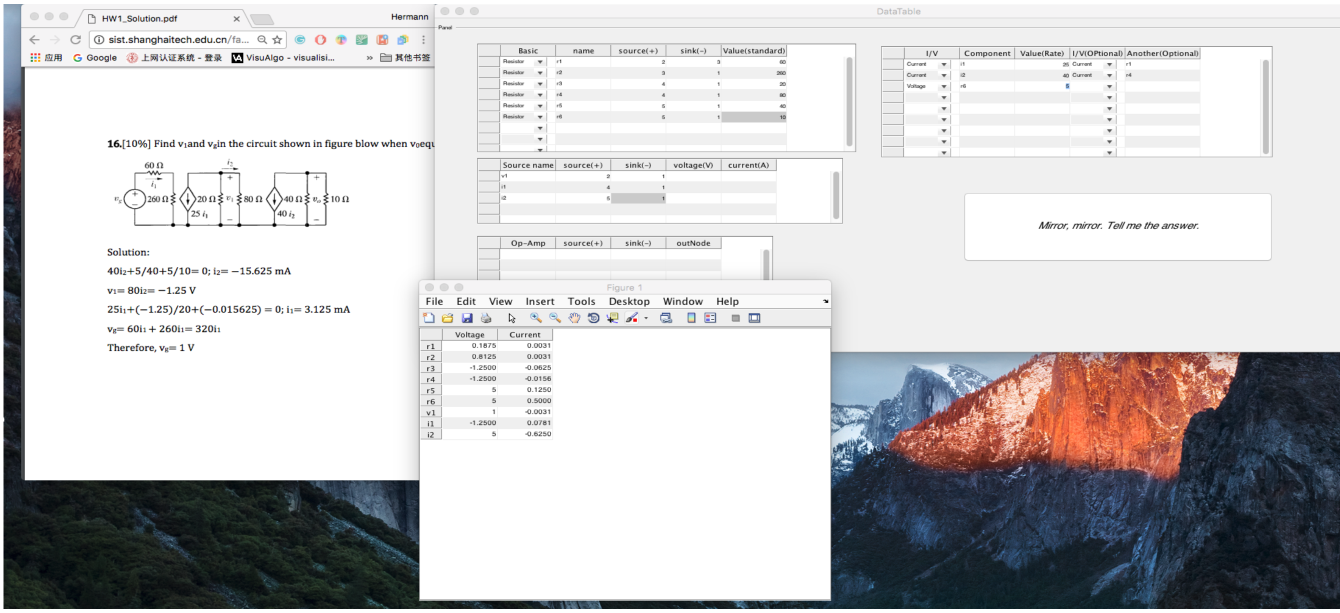


Figure 12: controlled source-3

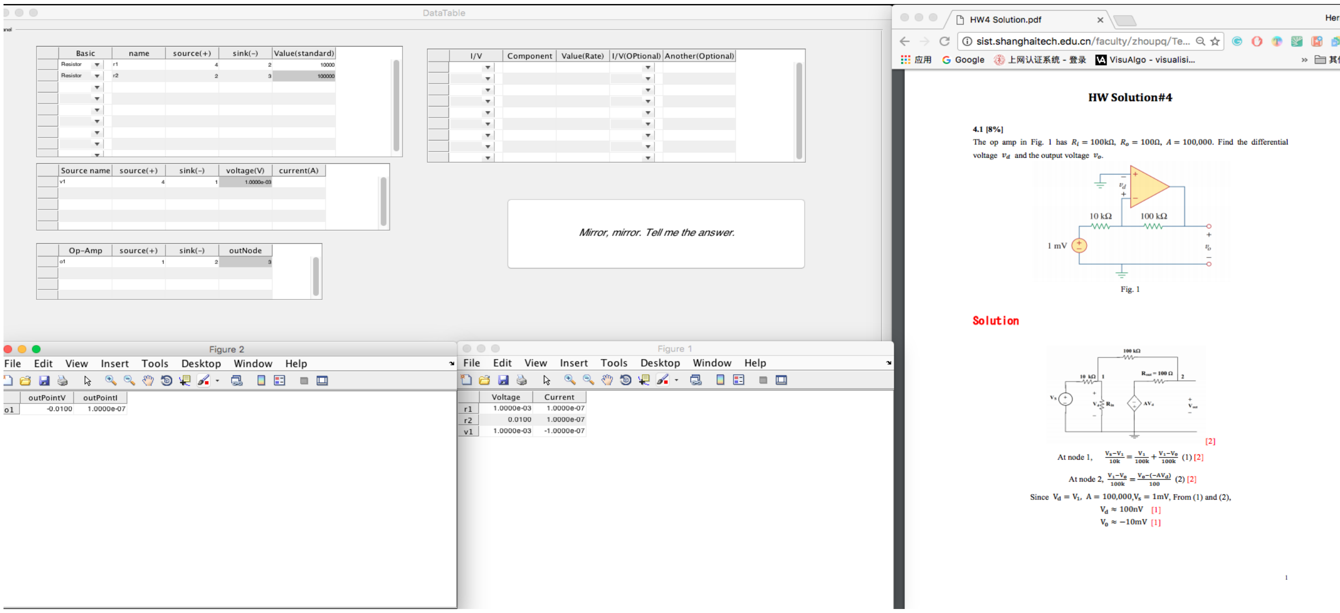


Figure 13: op-amp0

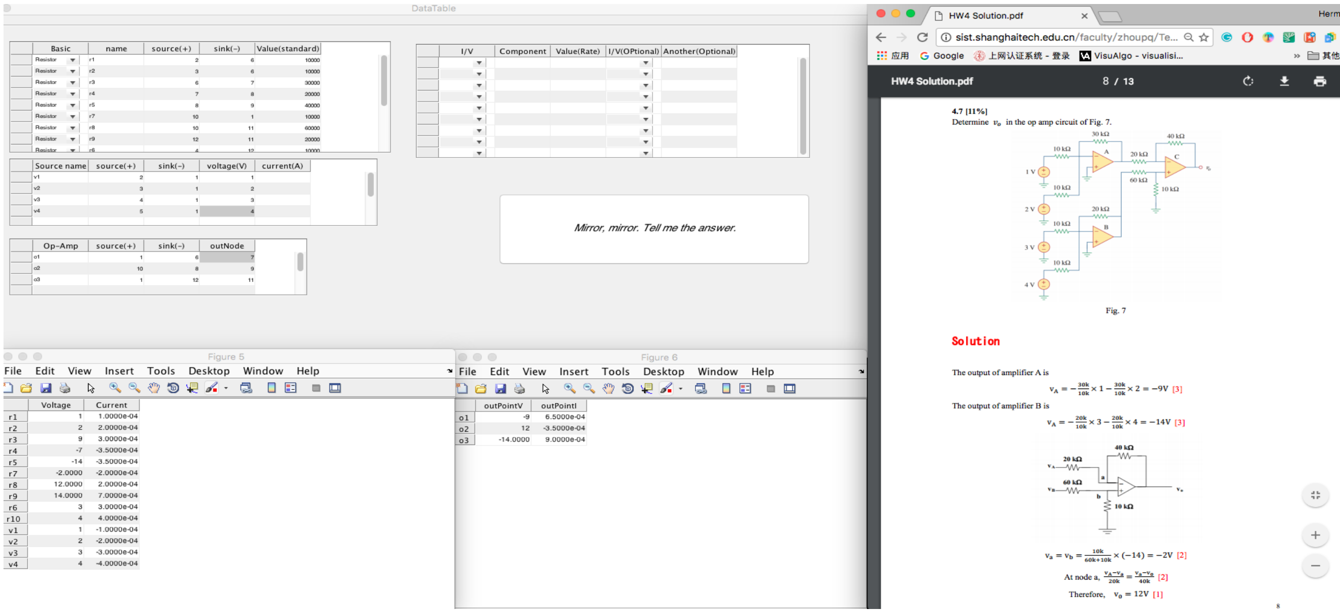


Figure 14: op-amp1

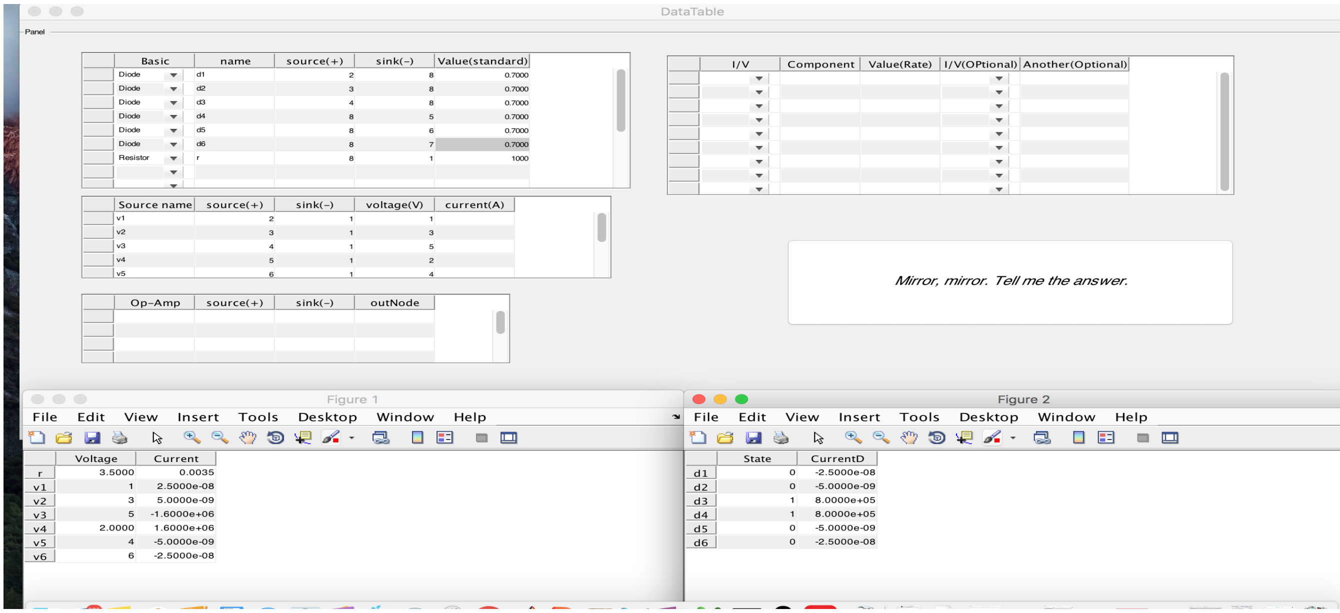


Figure 15: Diode- lab

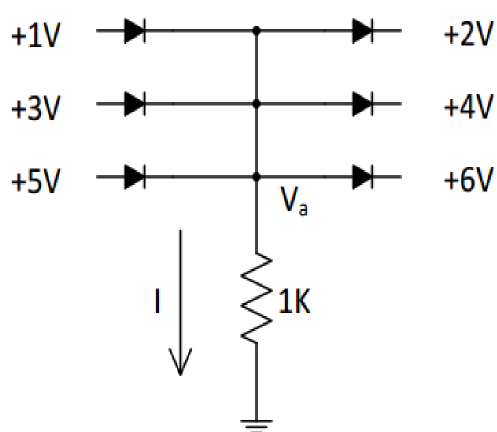


Figure 16: Diode

6 Iteration process

6.1 Passing Parameter in MATLAB

There's no exact pointer in MATLAB which cause a big trouble to my oop design. I have to set plenty of global variables to adopt it but finally face another trouble which cost lots of my time to reconstruct over and over again.

6.2 Omission in Share same ndoe

When two components with relation in voltage share same node, my program didn't record correct mathematical relation in matrix. It's error-prone since all materials mention it this way:

	v_j	v_k	i_{jk}	v_p	v_q	RHS
Node j			1			
Node k			-1			
Branch jk	1	-1		$-\mu$	μ	

Figure 17: Be careful that j may be the same as p

6.3 Diode

Diode is a troublemaker. I have to modify it to linear and iterate its discrete states to solve the linear system. Here are two cases which urge me to optimize model:

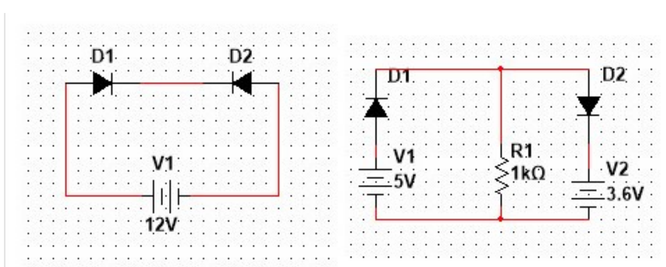


Figure 18: cases

The second can be abstract to be this:

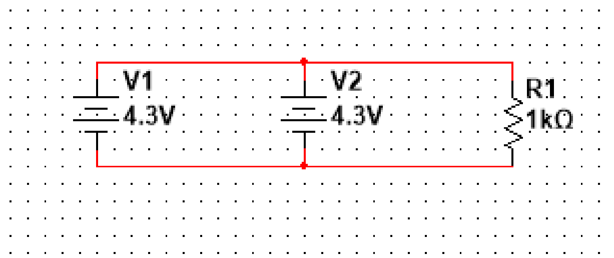


Figure 19: abstract

I used to major in chemistry so I have an intuition in how electron transfer in battery. Here is another similar question: whatll occur if two different voltage sources are in parallel. The answer in physical perspective tell us one of these two sources is on charge because of the thermodynamic theory but the distribution of current in branch is more complex because of chemical kinetics. It seems that two voltage sources share the same current which can be combined to 4.3 mA if the sources are the same. However, its an uncertain situation in mathematical perspective and even a meaningless condition because computer doesnt know how to distribute current in branch. When I run Multisim to simulate it, the program returned to me an error.

6.4 Phase Analysis

I have removed it because the conflict in interfaces and reserve only a default $\omega = 300$ to test capacitor and inductor.

6.5 Runing Speed

I make some simplication to avoid some trouble which effect less if the number of component is small. However, once I increase the number from 1000 to 10000, the time consuming increase five hundred times(reach to 557 seconds). This test indicate the inefficiency of dynamic allocating memory.

7 Personal Conclusion

7.1 Wang He

The project still has great room to be promoted. Because I am in charge of the test section, I had a deep impression that it is unfriendly to the users .
Because the user should enter all the information manually ,it is easy to input the wrong nodes ,especially for the big circuit.And annoyingly, if you are wrong ,you need to enter all information again which cost most of my time to test the project and check my inputs. And we choose the ideal model of operational amplifiers and diodes .There are more functions and different types of components can be added.
To find the bugs,I test some combinations of the six types of source, series resistors, parallel resistors, diodes, different operational amplifiers.Through testing the problems in the homework , exam and textbook,we find some indiscoverable bugs in our code.
Combining the knowledge of linear algebra and circuit is significative. In the information age, we can find the faster method to solve the problems.

7.2 Xie Zhiqiang

I'd like to save my time since I have communicated a lot with professors and spend lots of time to finish this report.
I'll work harder.