Michael Bliesath
ECE 1774

Project 1 – Simple Circuit Simulator

Project Overview

The Simple Circuit Solver is a program that solves the bus voltages and circuit current for a two-bus system. The two-bus system has a voltage source connected to Bus A (Bus 1). Bus A is connected to Bus B (Bus 2) through a resistor. Bus B is connected to a load, then the load is grounded. This configuration represents a basic series circuit and is shown schematically in Figure 1 below.

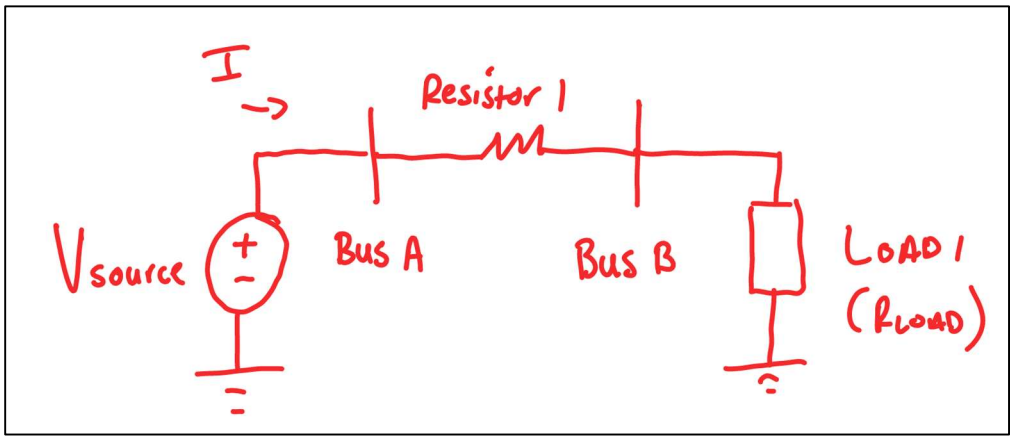

Figure 1: Simple Circuit Schematic

To solve this circuit automatically, there must be a hierarchy of base classes, executable classes, and solution logic to solve this circuit. This design allows individual components to be modeled independently while allowing the circuit to be assembled, solved, and analyzed by the program. The base classes represent the physical elements in the circuit like the resistors, loads, voltage sources, and buses. Each class encapsulates the parameters, validation rules, and calculations specific to that element. These base classes include Resistor.py, Load.py, VSource.py, and Bus.py.

As stated above, each of the base classes has their own logic and parameters associated with them that are used in the circuit. For Resistor.py, there are four inputs that include a string type name, bus object for the first bus, bus object for the second bus, a float type resistance. Then, there is one computed attribute which is a float type conductance. The conductance is calculated internally using the _calc_g() function. This function calculates conductance by taking the reciprocal of the resistance, $G = \frac{1}{R}$. Before the conductance is calculated, the resistance value is checked to be a positive value, otherwise it throws a ValueError because the conductance value would be undefined.

For Load.py, there are 2 variations for how this class works. The first variation is for milestone 2 of this project. This class has 5 inputs which are a string type name, bus object for

the first bus, a float type for real power, a float type for voltage, and a float type for resistance. Then, there is one computed attribute which is a float type conductance, which is computed the same as in Resistor.py. The second variation is after Milestone 2 for this project. This class has 4 inputs which are a string type name, bus object for the first bus, a float type for real power, a float type for voltage. Then, there are two computed attributes which are a float type for resistance, and a float type for conductance. Resistance is calculated using a variation of the power formula: $R = \frac{V^2}{P}$. Conductance is calculated the same way internally as before. Before the conductance is calculated, the resistance value is checked to be a positive value, otherwise it throws a ValueError because the conductance value would be undefined.

For VSource.py, there are 3 inputs which are a string type name, bus object for the first bus, and a float type for voltage. All these variables are defined to self and the voltage value is set to the input bus but assigning it through the bus voltage setter method (self.bus1.v = self.v). Therefore, the associated voltage is assigned to the bus, whereas without that there would be no connection to that voltage and the bus.

For Bus.py, there is one input which is a string type name. Then, there is one attribute which is a float type for voltage. The name is defined to self and the voltage value is set through a setter method (@v.setter). This setter method checks if the input voltage value is a valid value to the voltage variable, otherwise it raises a ValueError.

The Circuit.py class is an executable class that builds upon the base element classes (Resistor, Load, VSource, and Bus) to construct and manage the overall circuit. There is a single input which is a name of type string. The class contains a dictionary for buses with a string key, a list of resistors, a dictionary for loads with a string key, a voltage source object, and a float type for current. Aside from the circuit name, these attributes are initialized as empty or set to None and are populated as elements are added to the circuit. The add_bus method checks whether a bus with the given name already exists in the bus dictionary and, if not, adds that corresponding bus object. The add_resistor_element method checks if there are two bus objects from the two-bus name string inputs and verifies that the specified resistor name is unique and adds it to the resistor list. Similarly, the add_load_element method ensures that there is a bus object from the bus name string input, and checks if the load name is in the load dictionary. If the load name is not in the load dictionary, then it is added. The add_vsource_element method creates and assigns a voltage source to the circuit because there is only one in this circuit. The set_i method acts like a setter method and sets the current variable to the input. The print_nodal_voltage function just prints out the voltage values at the two buses. Lastly, the print_circuit_current function prints out the current of the circuit.

The Solution.py class is responsible for performing the circuit calculations and takes a Circuit object as its only input. This circuit object is stored internally and contains all elements needed to solve the circuit. The do_power_flow method validates that the circuit contains one voltage source, one resistor, one load, and two buses. Then, the method stores the resistance and

load values into temporary variables. The voltage at the first bus is stored in a variable (V_A). Next, the conductance values are stored into temporary variables and checked to ensure that they are positive values. The circuit current is solved by manipulating Ohm's law into the following equation: $I = \frac{V}{R}$. The resistance for this formula is calculated by adding the load conductance and resistor conductance together and taking the reciprocal of because they have an inverse relationship, which is shown by the following formula: $R_{total} = \frac{G_{load} * G_{resistor}}{G_{load} + G_{resistor}}$. Then, the circuit current is set through the set_i function from inheritance of the circuit class. Lastly, the voltage at the second bus is calculated through ohm's law and set to the second bus, $V = I * R$. The resistance value in this case is the reciprocal of the conductance on the load as Bus B is connected to the load, $R = \frac{1}{G_{load}}$.

Lastly, main.py file serves as the entry point for the circuit solver. In this script, a circuit object is created in addition to two bus objects. Both buses are added to the circuit object through the add_bus method. Next, a voltage source, resistor, and load are created and added to the circuit object. A solution object is created by calling the solution class with the circuit object as the input. The do_power_solver method is called in solution with the solution object which holds the solved circuit. Lastly, the print_nodal_voltages and print_circuit_current functions are called to display the bus voltages and circuit current.

Class Diagram

Michael Bliesath
ECE 1774

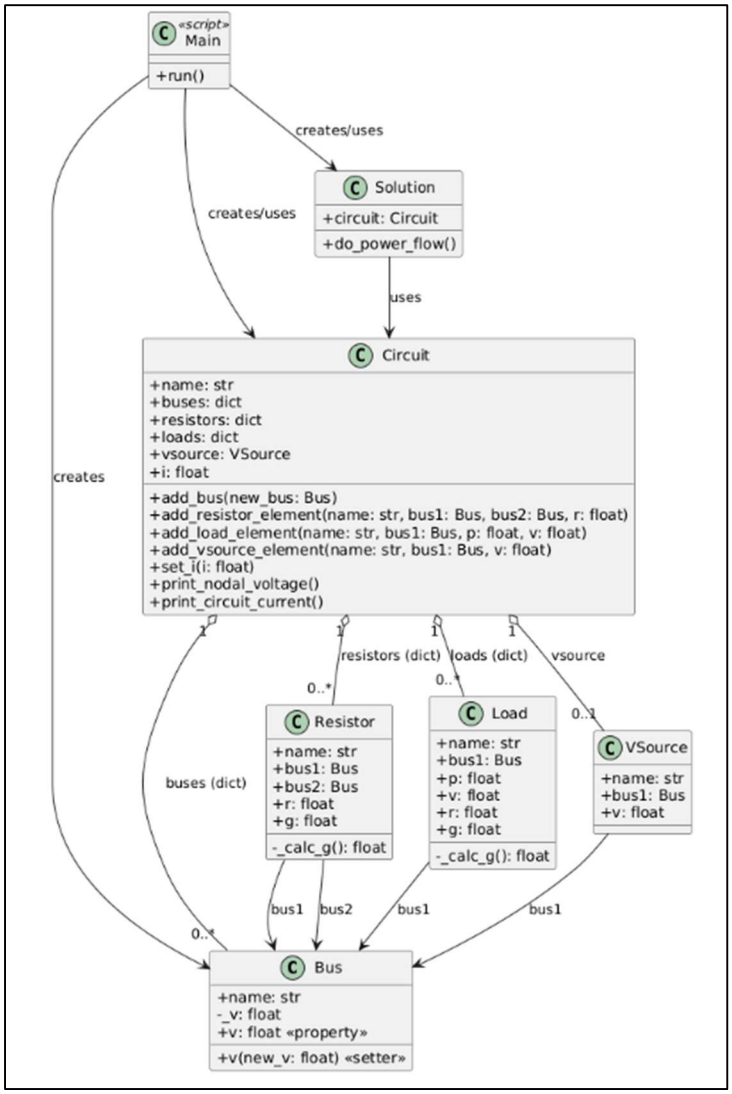

Figure 2: PlantUML Class Diagram

Relevant Equations

Table 1: Equations Used for Simple Circuit Solver

Conductance	$G = \frac{1}{R}$
Power Relationship	$R = \frac{V^2}{P}$
Ohm's Law	$V = I * R$
Resistors in Series	$R_{total} = R_1 + R_2$
Series Resistance Using Conductance	$R_{total} = \frac{G_{Load} * G_{Resistor}}{G_{Load} + G_{Resistor}}$

Michael Bliesath
ECE 1774

Kirchoff's Current Law at Bus A	$V_A = I * R_{AB} + I * R_{Load}$
Kirchoff's Loop Law for the Circuit	$V_A - I * R_{AB} - V_B - I * R_{Load} = 0$

Example Case

The example case considered in this project corresponds to the circuit specified in Milestone 4. This circuit consists of two buses, Bus A and Bus B. A voltage source, V_A, is connected at Bus A and is set to 100V. Next, a resistor, R_{AB}, is created that connects Bus A and Bus B and has a resistance of 5Ω. The last element of the circuit is the load which is connected to Bus B and has a real power of 2000W and a nominal voltage of 100V.

To determine the voltages at Bus A and Bus B, the current must be solved first. The voltage at Bus A is just the voltage source value because it is the only circuit element connected directly to Bus A that provides voltage. Therefore, the voltage at Bus A is just 100V. Since the voltage at Bus A is known, the current can be solved since this circuit is all in series so the current remains the same throughout all the elements. To solve for the current, Kirchoff's current law can be used at the Bus A junction which gives the following formula: $V_A = I * R_{AB} + I * R_{Load}$. This formula can be rearranged to the following formula to solve for the current: $I = \frac{V_A}{R_{AB}+R_{Load}}$. The voltage value at V_A is just the source voltage value as explained above, which is 100V. The resistance at Resistor R_{AB} is 5Ω, however the resistance at the load is unknown. The power formula can be used to solve resistance: $R = \frac{V^2}{P}$. Inserting the nominal voltage at the load, 100V, and the real power of the load, 2000W, the resistance value of the load is also 5Ω.

Therefore, the current can be solved for which gives a value of 10A. To solve the voltage at Bus B, Kirchoff's loop law must occur which gives the following formula: $V_A - I * R_{AB} - V_B - I * R_{Load} = 0$. Next, the formula is rearranged to solve the voltage at Bus B as such $V_B = V_A - I * R_{AB} - I * R_{Load}$. All these values were given or solved previously, so plugging them in gets a voltage value of 50V. This is correct because if Kirchoff's current law is done at Bus B, the resistance to the load multiplied by the current would just be 10A times 5Ω which gives 50V.

Solving this example case using the solver uses a similar process. The following steps are describing the script done below in figure 3. There is a circuit object created. Next, there are 2 bus objects created and added to the circuit object. Following this, a voltage source element is created that is connected to Bus A, has a voltage value of 100V assigned to it, and is named V_A. Then, a resistor element is created that connects Bus A and Bus B together that has a resistance of 5Ω and is named R_{AB}. Subsequently, a load element is created and attached to Bus B, has a nominal voltage value of 100V, 2000W of real power, and is named L_B. All these elements and values match the case explained above.

Next, a solution object is created from the circuit object. This solution object uses its do_power_flow function which runs the solver that does the same process explained above.

Lastly, both print functions are called to the circuit object which is now solved and prints all the bus voltage values and circuit current value. These values are shown below in figure 4, which match the values solved manually above. Therefore, this solver successfully solves the circuit and can simulate similar circuits with different values.

```
if __name__ == "__main__":
    c = Circuit("SimpleCircuit")

    a = Bus("A")
    b = Bus("B")

    c.add_bus("A")
    c.add_bus("B")

    c.add_vsource_element( name: "Va",  bus1_name: "A",  v: 100.0)

    c.add_resistor_element( name: "Rab",  bus1_name: "A",  bus2_name: "B",  r: 5.0)

    c.add_load_element( name: "Lb",  bus1_name: "B",  p: 2000.0,  v: 100.0)

    solution = Solution(c)
    solution.do_power_flow()

    c.print_nodal_voltage()
    c.print_circuit_current()
```

Figure 3: Example Case Parameters

```
"C:\Users\Michael Bliesath\PycharmProjec
Bus A Voltage: 100.0 V
Bus B Voltage: 50.00000000000001 V
Circuit Current: 10.000000000000002 A

Process finished with exit code 0
```

Figure 4: Example Case Solution