# EAGLECAD Circuit Simulation Tutorial

*ECE 445*
*University of Illinois at Urbana-Champaign*
*Department of Electrical and Computer Engineering*

## Instructions

This documentation includes explanations of EagleCAD's integrated SPICE software and specific pointers for the Circuit Simulation assignment. The scope of this document is limited to covering the essentials for completing the associated assignment. This guide is heavily derived from Autodesk's SPICE Simulation blog series [1]-[6], and it is highly recommended to read through these pages for additional details about using SPICE within Eagle.

## Overview

What is SPICE? SPICE is a computer simulation developed at the University of California, Berkeley. It's one of the most widely used simulation programs that allows engineers to simulate the behavior of circuits. Why bother with simulating though? By testing how a circuit will perform in action, you can catch errors early in the design process before they manifest in your prototypes.

SPICE simulation makes this happen through the use of SPICE models and a netlist. The netlist defines how pins are connected on your schematic. Models contain text descriptions of a component's values, which can then be used by the SPICE engine to simulate behavior mathematically.

Models for simple components can be simple one-line descriptions. Complex multi-part components will often have models with hundreds of lines of information. Unless you plan to use a library in Eagle that already has spice compatible parts, then you'll need to download SPICE models yourself. Thankfully there are plenty of component manufacturers who provide SPICE models to download on their websites.

Remember when searching for SPICE models that there's a difference between PSpice and SPICE. PSpice is a proprietary simulator owned by Cadence, and many PSpice models are not compatible with standard SPICE simulators. To avoid any confusion, always stick to downloading SPICE3 (the latest 3f5) or SPICE2 models, as they are all versions of the good old Berkeley SPICE.

SPICE is fully integrated into Autodesk EAGLE 8.4, and includes four simulation types:

- **Transient Analysis**. This method will simulate how a circuit performs over time as current travels through every component in your circuit.

- **DC Sweep**. This method analyzes the relationship between voltage and current over a given range for a DC input source.
- **AC Sweep**. This method analyzes changes in voltage and current in a given frequency range for an AC input source.
- **Operating Point**. This method analyzes the voltage and current for one or multiple components in a circuit at a single point in time. You can even extract and analyze specific component parameters.

This tutorial will focus only on Operating Point and DC Sweep, as these are the simulations required for the Circuit Simulation Assignment. However, it is recommended that you learn about all simulation types, as that will aid in your analysis of your custom circuits for your projects. Eagle comes with several pre-configured circuit examples that will allow you to quickly test out the SPICE simulator without needing to configure settings. To find these examples, open your **Control Panel**, expand the **Project** folder, and look for the **ngspice** folder in the **examples** section.

Eagle includes an Ngspice managed library which contains a set of parts that are simulation ready. There's everything from basic passive components like resistors and diodes to more advanced parts like BJT transistors, voltage controlled sources, and more. To check out all of the parts in this library, open your Add dialog and look for the *ngspice-simulation.lbr* listing. One thing to keep in mind, if you don't use parts from this SPICE compatible library, then you will need to download and map SPICE models to your schematic symbols. This process is outside the scope of this guide but is covered in detail in Autodesk's blog [5][6].

## DC Operating Point Analysis

One of the most fundamental SPICE simulations is DC Operating Point Analysis. This method will allow you to analyze the behavior of a circuit when a DC voltage or current is applied. In our example project, we'll be verifying the expected current from a voltage source to ground. This simulation might sound simple on the outside, but it's a great way to learn the basics of how SPICE simulation works in Eagle.

We will start by creating a new schematic, and drawing the circuit shown in Figure 1. Since we plan to run simulations on this circuit, we must use components that have SPICE models attached to them. We will use Eagle's inbuilt **ngspice-simulation** library (Figure 2) to select the voltage sources, resistors, and ground components. Once you've added the components and nets, set the values and name the nets to match Figure 1. Using KVL, we can see that the current from node A to ground should be 0.1 mA. We will use the DC Operating Point simulation to verify that this is true.

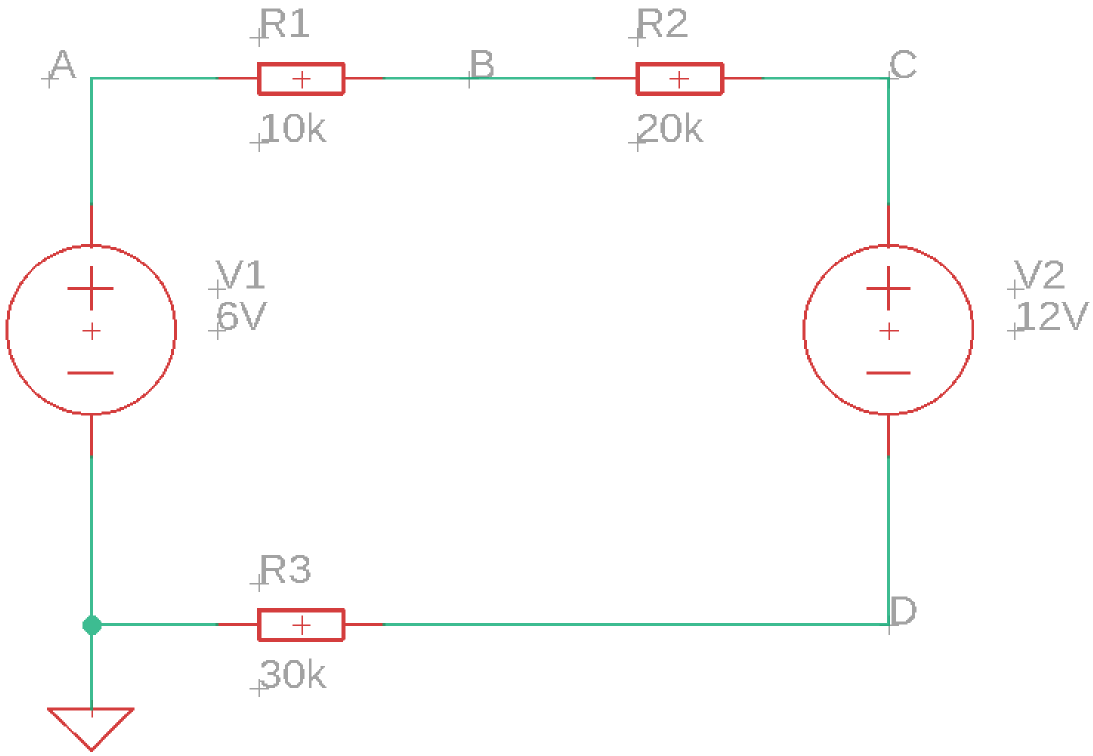
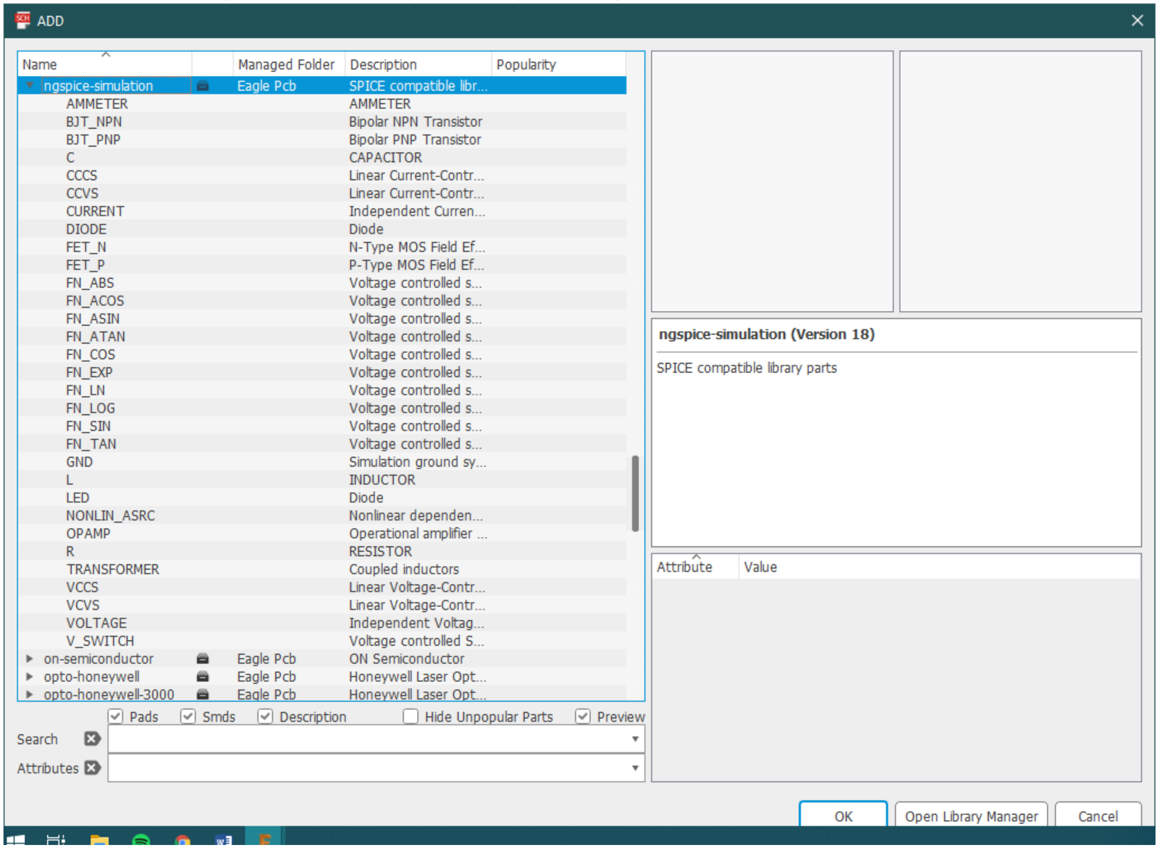


*Figure 1: Circuit for DC Operating Point Analysis simulation*



| Name | Managed Folder | Description | Popularity |
|---|---|---|---|
| ngspice-simulation | Eagle Pcb | SPICE compatible libr... | |
| AMMETER | | AMMETER | |
| BJT_NPN | | Bipolar NPN Transistor | |
| BJT_PNP | | Bipolar PNP Transistor | |
| C | | CAPACITOR | |
| CCCS | | Linear Current-Contr... | |
| CCVS | | Linear Current-Contr... | |
| CURRENT | | Independent Curren... | |
| DIODE | | Diode | |
| FET_N | | N-Type MOS Field Ef... | |
| FET_P | | P-Type MOS Field Ef... | |
| FN_ABS | | Voltage controlled s... | |
| FN_ACOS | | Voltage controlled s... | |
| FN_ASIN | | Voltage controlled s... | |
| FN_ATAN | | Voltage controlled s... | |
| FN_COS | | Voltage controlled s... | |
| FN_EXP | | Voltage controlled s... | |
| FN_LN | | Voltage controlled s... | |
| FN_LOG | | Voltage controlled s... | |
| FN_SIN | | Voltage controlled s... | |
| FN_TAN | | Voltage controlled s... | |
| GND | | Simulation ground sy... | |
| L | | INDUCTOR | |
| LED | | Diode | |
| NONLIN_ASRC | | Nonlinear dependen... | |
| OPAMP | | Operational amplifier ... | |
| R | | RESISTOR | |
| TRANSFORMER | | Coupled inductors | |
| VCCS | | Linear Voltage-Contr... | |
| VCVS | | Linear Voltage-Contr... | |
| VOLTAGE | | Independent Voltag... | |
| V_SWITCH | | Voltage controlled S... | |
| on-semiconductor | Eagle Pcb | ON Semiconductor | |
| opto-honeywell | Eagle Pcb | Honeywell Laser Opt... | |
| opto-honeywell-3000 | Eagle Pcb | Honeywell Laser Opt... | |

☑ Pads ☑ Smds ☑ Description □ Hide Unpopular Parts ☑ Preview

Search

Attributes

**ngspice-simulation (Version 18)**

SPICE compatible library parts

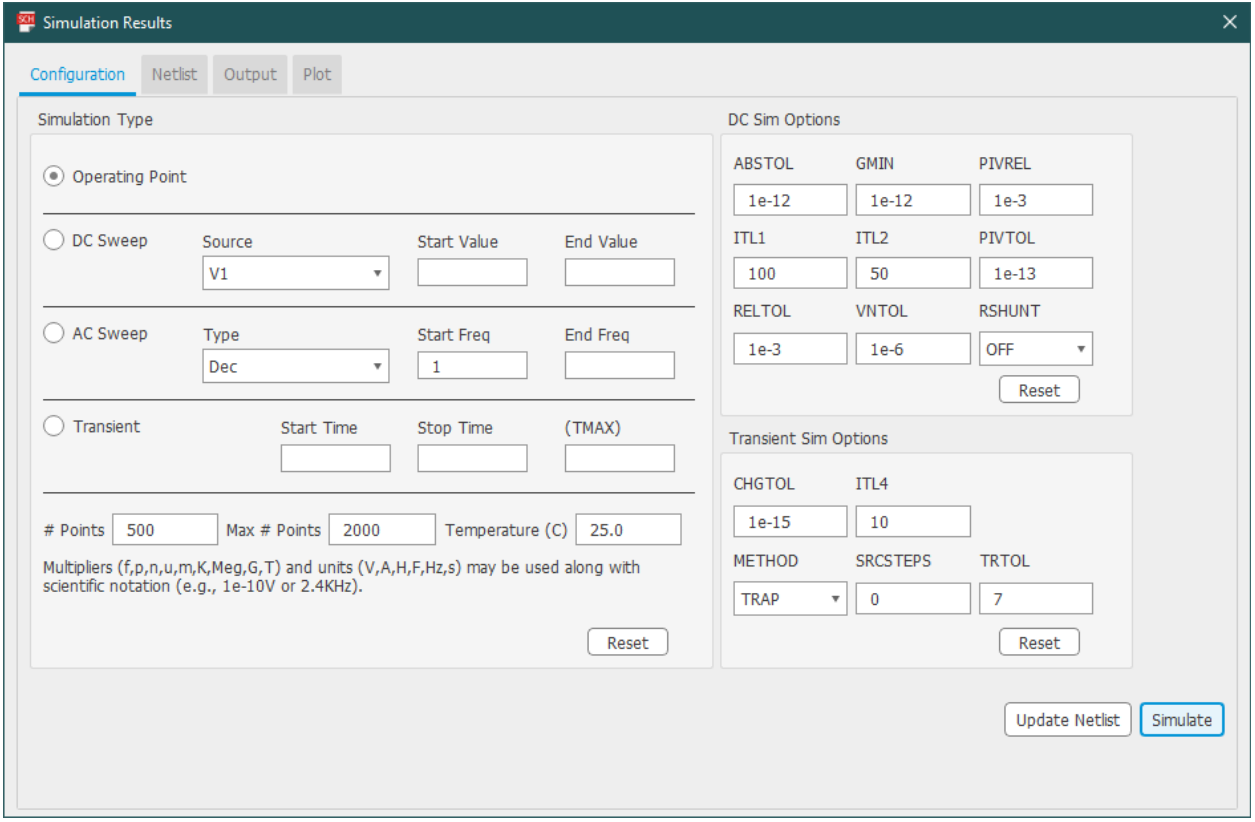
| Attribute | Value |
|---|---|

*Figure 2: Eagle's ngspice-simulation library*

Every SPICE simulation is run from the **Simulation dialog**. You can open this in one of two ways. Either select the **Simulation icon** at the top of your interface or enter "sim" in the **command line** and press **Enter**. You'll notice four tabs when you open the simulation dialog, let's walk through each before running our operating point analysis.

The first tab (Figure 3) is where you'll set the simulation type. The left-hand side contains all of the available simulation options. On the right, you have your **DC Sim** and **Transient Sim** options. You likely won't ever need to change these values unless your simulation runs into converging or timestamp issues. These and other common SPICE related issues can be resolved by tweaking the DC sim and Transient Sim options. For this guide we'll leave all these values at default. If you change one by mistake just press **reset** to start fresh.

An essential button on this tab is the **Update Netlist button**. Whenever you make a change to a symbol value or name on your schematic, pressing this button will update your SPICE netlist configuration. There's also the **Simulate button**, which we'll be working with later.



Simulation Results

Configuration | Netlist | Output | Plot

Simulation Type

● Operating Point

○ DC Sweep — Source: V1 — Start Value: — End Value:

○ AC Sweep — Type: Dec — Start Freq: 1 — End Freq:

○ Transient — Start Time: — Stop Time: — (TMAX):

# Points: 500 — Max # Points: 2000 — Temperature (C): 25.0

Multipliers (f,p,n,u,m,K,Meg,G,T) and units (V,A,H,F,Hz,s) may be used along with scientific notation (e.g., 1e-10V or 2.4KHz).

DC Sim Options

| ABSTOL | GMIN | PIVREL |
|---|---|---|
| 1e-12 | 1e-12 | 1e-3 |
| **ITL1** | **ITL2** | **PIVTOL** |
| 100 | 50 | 1e-13 |
| **RELTOL** | **VNTOL** | **RSHUNT** |
| 1e-3 | 1e-6 | OFF |

Transient Sim Options

| CHGTOL | ITL4 | |
|---|---|---|
| 1e-15 | 10 | |
| **METHOD** | **SRCSTEPS** | **TRTOL** |
| TRAP | 0 | 7 |

*Figure 3: Simulation Dialog*

The **Netlist tab** (Figure 4) displays a raw text format of your netlist configuration, which includes your simulation settings, devices, models, and more. The netlist is a text-based representation of your visual schematic. If you change values here, it won't change anything on your schematic.

Under the **Options heading** are all of the values that you set in the Configuration tab. You can also change these values directly here if you'd prefer. Look under the **Devices heading**, and you'll see all of the parts connected in your schematic. The format goes like this:

*Type_Designator, Connection 1, …, Connection N, Value*

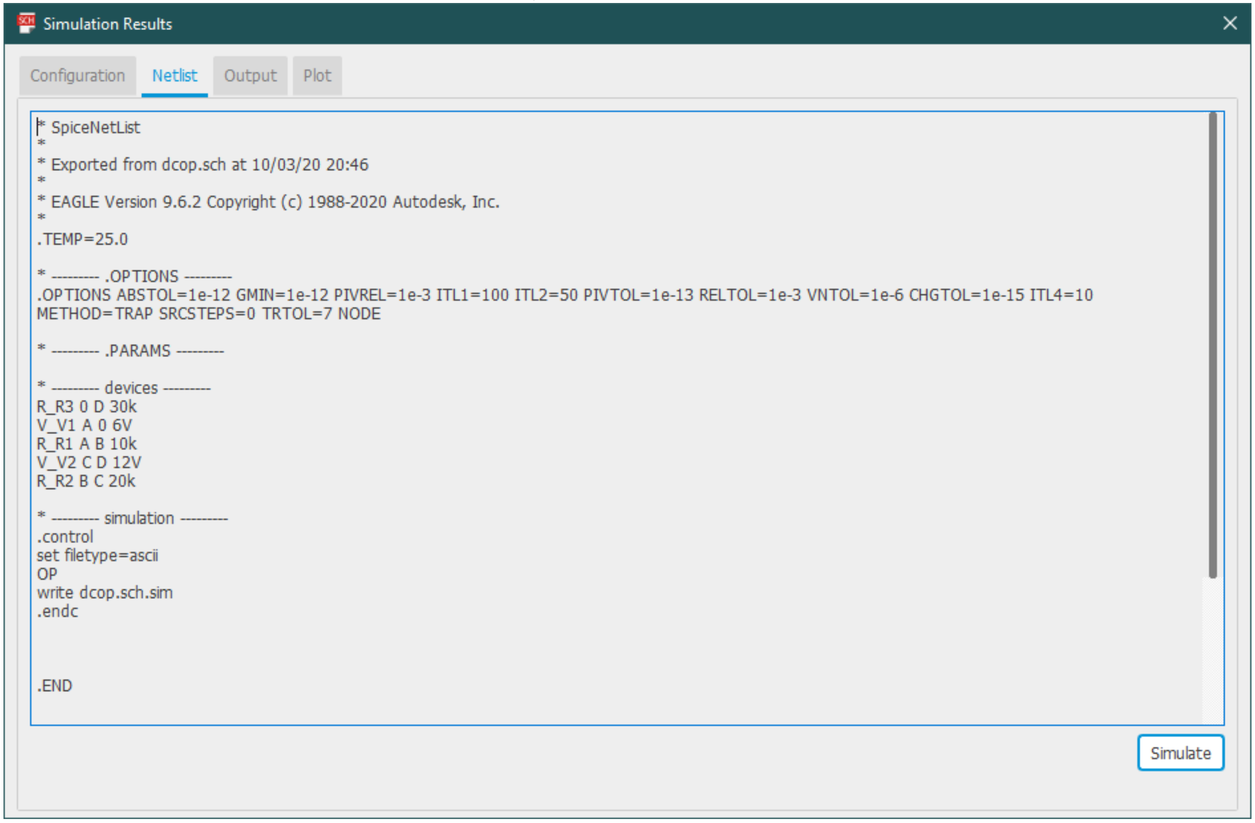
For example,

*R_R1 A B 10k*

This device is a resistor, **R**, with reference designator **R1**, pins connected to nets **A** and **B**, with a resistance value of **10k**.

You can quickly copy and paste information to/from your netlist and also make quick changes to parameters. For example, we could change our R1 resistor to 12k, or our V1 voltage source to 3.5V, and then immediately run our simulation based on this edited netlist. This is a great option if you want to quickly test values and compare results without having to change your schematic.

One important note, when you select the **Update Netlist button** on your **Configuration tab** this will recreate the netlist with parameters from your schematic. Use this button if you ever tweak values and want to revert to your default netlist values.



```
* SpiceNetList
*
* Exported from dcop.sch at 10/03/20 20:46
*
* EAGLE Version 9.6.2 Copyright (c) 1988-2020 Autodesk, Inc.
*
.TEMP=25.0

* --------- .OPTIONS ---------
.OPTIONS ABSTOL=1e-12 GMIN=1e-12 PIVREL=1e-3 ITL1=100 ITL2=50 PIVTOL=1e-13 RELTOL=1e-3 VNTOL=1e-6 CHGTOL=1e-15 ITL4=10
METHOD=TRAP SRCSTEPS=0 TRTOL=7 NODE

* --------- .PARAMS ---------

* --------- devices ---------
R_R3 0 D 30k
V_V1 A 0 6V
R_R1 A B 10k
V_V2 C D 12V
R_R2 B C 20k

* --------- simulation ---------
.control
set filetype=ascii
OP
write dcop.sch.sim
.endc


.END
```

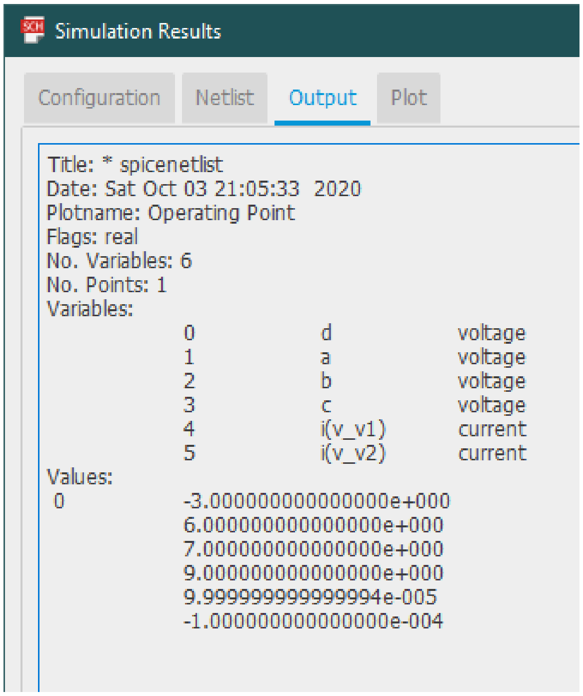
*Figure 4: Netlist*

These last two tabs are where all of your simulation outputs will display. Raw text results will display in the **Simulator Output tab**, and the **Plot tab** will provide a visual graph for transient analysis methods.

Since our operating point analysis is only measuring values at a single point in time results will only be displayed in the Simulator Output tab.

The Simulator Output tab is also where you can see any errors with your simulation. This tab is important to review for simulation specific errors. For example, changing the value on a symbol might work for EAGLE, but not the SPICE simulator. When a simulation is run, EAGLE will check all of your component values and alert you about any simulation errors.

Now it's time to run the simulation. Open the **Simulation dialog** and go to the **Configuration tab**. Under **Simulation Type**, select **Operating Point**, set **# of points** to **500**, and set **Temperature** to **25**. If you made any changes to the netlist in the Netlist tab earlier, be sure to click the Update Netlist button. Then click on **Simulate**.



```
Title: * spicenetlist
Date: Sat Oct 03 21:05:33  2020
Plotname: Operating Point
Flags: real
No. Variables: 6
No. Points: 1
Variables:
                0           d           voltage
                1           a           voltage
                2           b           voltage
                3           c           voltage
                4           i(v_v1)     current
                5           i(v_v2)     current
Values:
 0              -3.000000000000000e+000
                6.000000000000000e+000
                7.000000000000000e+000
                9.000000000000000e+000
                9.999999999999994e-005
                -1.000000000000000e-004
```

*Figure 5: DC Operating Point Analysis simulation output*

The Output tab should automatically open (Figure 5), with the results of the simulation. Here you can see a breakdown of the voltage and current as it travels through each node in the circuit. From my simulation, the current through V1 is the fifth row, although this might vary for you. We can see that the value for this variable is 9.999999999999994e-005, which is within a rounding error of 0.1 mA, the value we expected to see. Note that positive current is defined as travelling from the positive terminal to the negative terminal across the voltage source.

If you close the Simulation dialog, you'll also see simulation results displayed on your schematic in blue (Figure 6). Here you can also see that we're getting 0.1 mA of current through node A to ground. You can toggle the display of simulation results on your schematic by selecting the **Sim Results Toggle** 🔍 **icon** at the top of your interface.

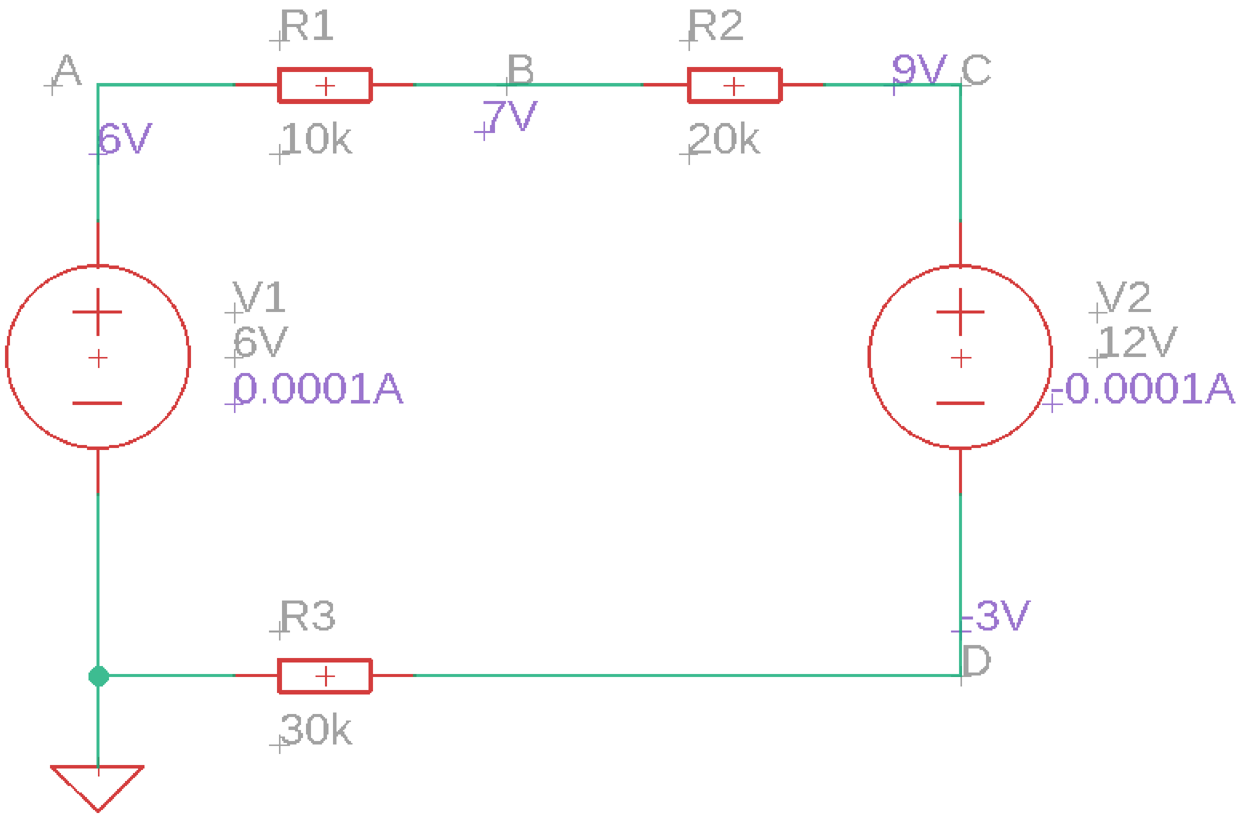


*Figure 6: Simulation results added to schematic*

## DC Sweep Analysis

DC Sweep simulations essentially allow you to run multiple Operating Point simulations for a defined range of voltages. We will reuse our circuit from the previous section to explore how to work with this type of simulation. We will sweep V1 over a range of voltages to see how the current through V1 changes. Open the **Simulation dialog**, and select **DC Sweep** for the **Simulation Type**. Select V1 for the **Source**, set **Start Value** to **0**, and **End Value** to **20**. The Start and End Values are the ranges for the voltage sweep. Again, set **# of points** to **500**, and set **Temperature** to **25**. Leave all the **DC Sim Options** at their defaults, using the **Reset** button if necessary. Clicking on the **Simulate** button will bring up the **Plot tab** with the results of the sweep (Figure 7).

The plot looks confusing at first glance but is actually easy to navigate once you understand how it is organized. The voltage sweep range is on the X-axis, voltages are on the left-side Y-axis, and currents are on the right-side Y-axis. If you move your mouse from left to right across the plot, it will follow all the lines and update the values on the right side of the window. Following a specific line with the cursor will highlight the corresponding value on the right.

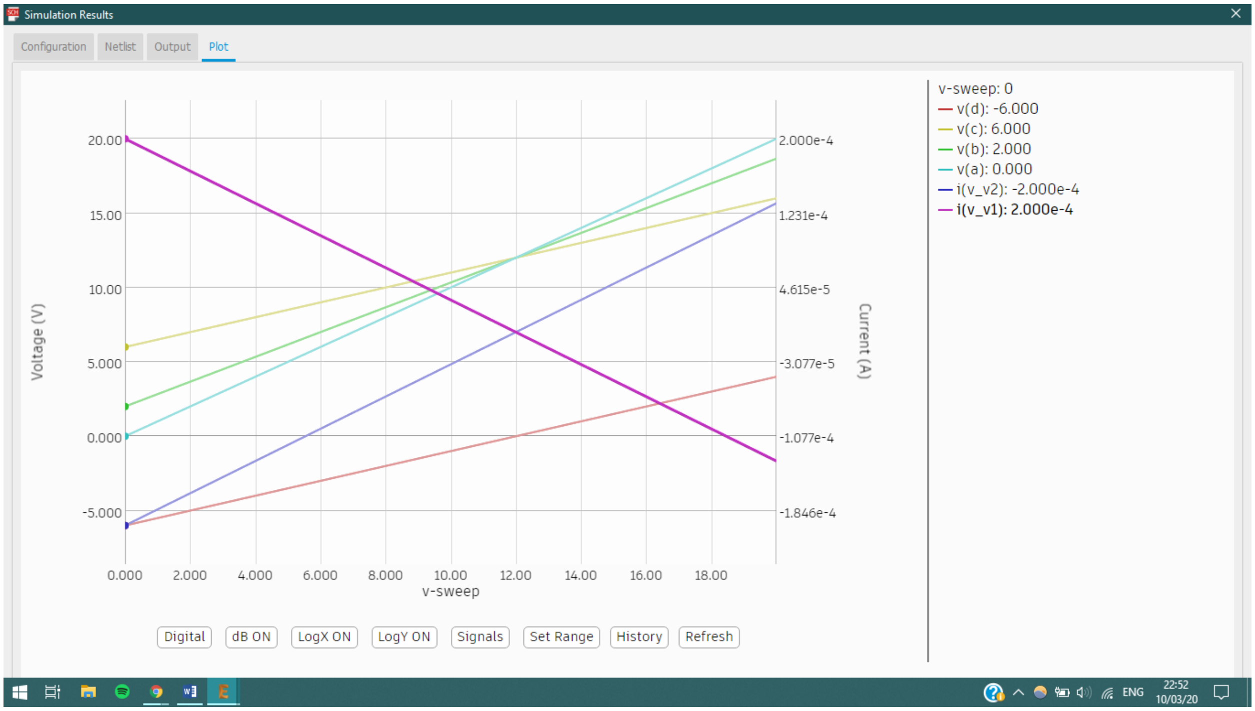


*Figure 7: Output plot for DC Sweep simulation*

You can zoom in on any part of the graph by left-clicking and dragging over a specific area. Double-clicking will restore the default view. There are also several buttons underneath the plot.

- **Digital**: toggles between analog and digital readings of values. Useful if you are simulating a ADC or DAC circuit
- **dB ON**: converts voltage output to a dB scale
- **LogX/LogY ON**: converts the scale of the corresponding axis from linear to log
- **Signals**: allows you to toggle which signals to display on the plot
- **Set Range**: allows you to control the limits of the three axes
- **History**: allows you to save up to 5 plots
- **Refresh**: resets all settings to the default

## Voltage Probes and Multipliers

The last tool for SPICE simulation needed for the assignment is the Voltage Probe. Close the Simulation Dialog and click on the **Voltage Probe** **icon**. Click on net A, and then net C. You should see a new Voltage Probe **V(A)** added to the schematic (Figure 8). Rerun the simulation and see which signals are plotted. You will notice that now none of the net voltages are reported, unlike before. Instead, the only voltage reported is V(A), the custom voltage probe we placed on the circuit. Whenever one or more voltage probes are used, the simulation will always only report the voltages from the probes, without reporting all net voltages. This makes voltage probes a useful way to look at specific voltages of interest without cluttering the plot with unnecessary information.

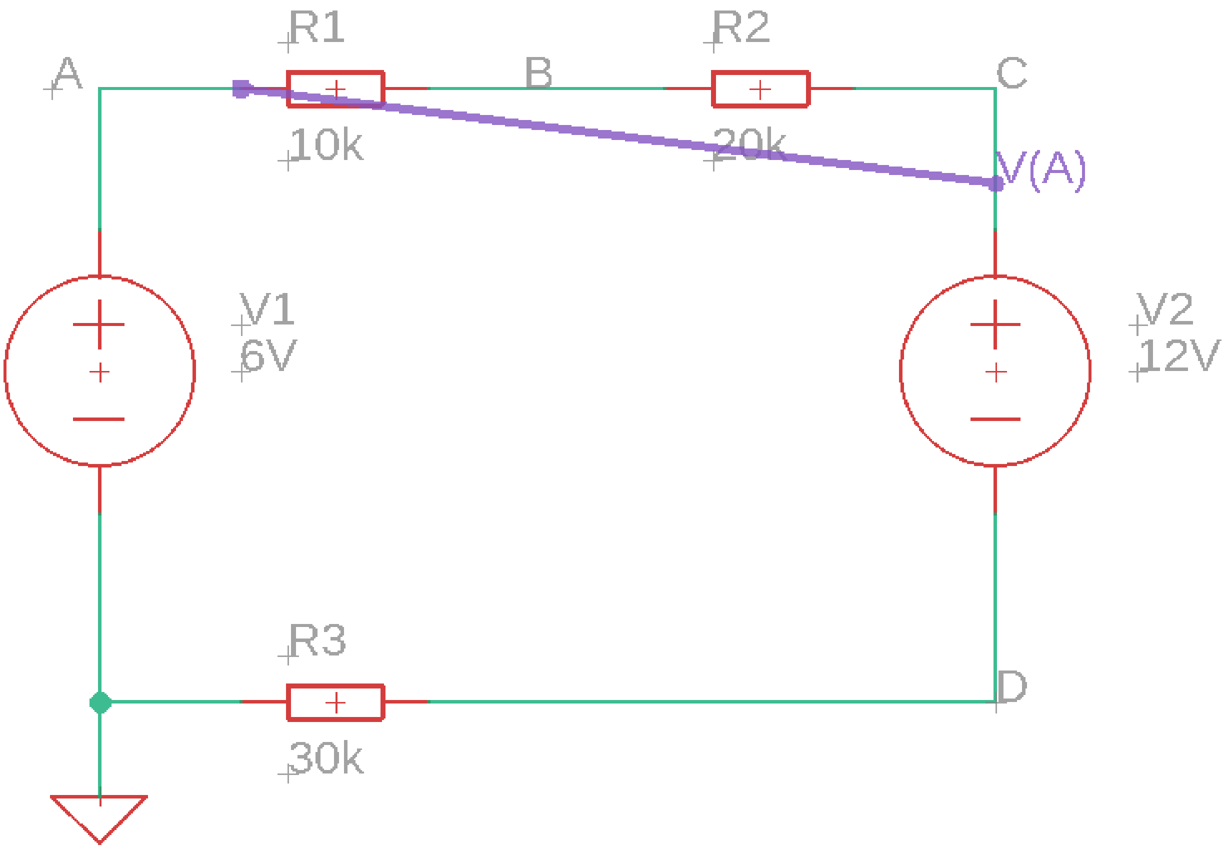


*Figure 8: Voltage Probe from net A to net C*

Another useful tool is adding multipliers to the simulated components. For example, say we wanted to look at the current from ground to net A through V1 (the more intuitive way to think about the current through a voltage supply). How can we tell SPICE to negate the calculated value so we don't have to? We can edit the **Netlist** to tell SPICE to do simple math operations. In the **Simulation** section in the Netlist, there is a line that reads

*write sim_tut.sch.sim V(A) I(V_V2) I(V_V1)*

This is the code that tells SPICE what information to output from the simulation. Since we want to negate the current through V1, we can simply edit the line to look like this

*write sim_tut.sch.sim V(A) I(V_V2) I(V_V1)*-1*

Clicking on **Simulate** from the Netlist tab will bring up the plot with the current negated to match our intuitive idea of how to measure current. Most basic math operations are supported in this editor, including basic arithmetic, exponentiation, and trig functions. Don't forget that clicking **Update Netlist** on the **Configuration tab** will overwrite any changes you have made in the Netlist.

## References

[1] S. Sattel, “SPICE Simulation Basics Part 1: Getting Started: EAGLE: Blog,” *Eagle Blog*, 24-Oct-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-1. [Accessed: 14-Sep-2020].
[2] S. Sattel, “How-To SPICE Simulation Operating Point: EAGLE: Blog,” *Eagle Blog*, 23-Nov-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-2-operating-point-analysis/. [Accessed: 14-Sep-2020].
[3] S. Sattel, “How-To SPICE Simulation DC/AC Sweep: EAGLE: Blog,” *Eagle Blog*, 07-Nov-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-3-dc-ac-sweep-analysis/. [Accessed: 14-Sep-2020].
[4] S. Sattel, “Transient Analysis & SPICE Model Mapping: EAGLE: Blog,” *Eagle Blog*, 11-Nov-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-4-transient-analysis-spice-model-mapping/. [Accessed: 14-Sep-2020].
[5] S. Sattel, “Schematic to SPICE Model Mapping: EAGLE: Blog,” *Eagle Blog*, 07-Dec-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-5-schematic-spice-mapping/. [Accessed: 14-Sep-2020].
[6] S. Sattel, “Mapping SPICE Compatible Libraries: EAGLE: Blog,” *Eagle Blog*, 07-Dec-2017. [Online]. Available: https://www.autodesk.com/products/eagle/blog/spice-simulation-part-6-make-eagle-libraries-spice-ready/. [Accessed: 14-Sep-2020].