

UNIVERSITY OF CALIFORNIA
College of Engineering
Department of Electrical Engineering and Computer Sciences

HSPICE Tutorial

1. Objective

The objective of this session is to give initial exposure to the software environment that will be used in the lab session throughout the semester.

HSPICE (a circuit simulator) and WaveView Analyzer (a waveform viewer) will be used to execute some of the procedures that are necessary in many lab assignments, in addition to your project, during the course.

2. Set up the simulation environment

- a. Login the computer in Cory 125 using your WINDOWS instructional account. You may want to change the password using the command listed on the instructional account sheet for both WINDOWS and UNIX for future use. In our lab, we will use LINUX environment for running HSPICE.

Connect to a LINUX server:

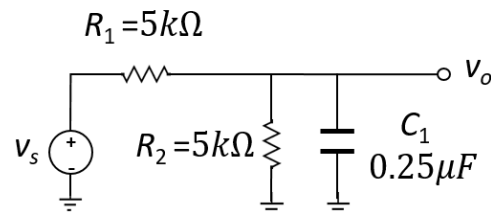
Software PUTTY and Xming is used to connect to a LINUX server. First open the Xming software, then open PUTTY and type the server name “hpse-9.eecs.berkeley.edu” as the Host name. **Here remember to click SSH→X11 and enable X11 forwarding.**

Login using your instructional account:

After first time login, you can change the password using command: “ssh update” for UNIX account or ‘CTRL+ALT+DEL’ then click change password for WINDOWS account.

- b. Create and direct to the workspace folder named ‘EE105’. Open your favorite text editor and type in the netlist of the circuit shown in Figure T1.1. You may want to do this as follows:

```
> mkdir EE105
> cd EE105
> vim your_ckt.sp
```



$$v_s = \begin{cases} 0 & 0 \sim 5ms \\ 0 \rightarrow 5V & 5ms \sim 5.1ms \\ 5V & 5.1ms \sim 10ms \end{cases}$$

Figure T1.1

- c. You may use the netlist below for Figure T2.1.

```
EE105 FALL18 SAMPLE LP FILTER CIRCUIT
vs vs gnd pwl(0ms 0V 5ms 0V 5.1ms 5V 10ms 5V)
r1 vs v0 5k
r2 v0 gnd 5k
c1 v0 gnd 0.25uF
.tran 0.001ms 10ms
.options post=2 nomod
.end
```

*Options, such as output file and simulation report format, are controlled by proper commands listed after ‘.options’. ‘post=2’ instructs the simulator to generate a WaveView-readable file such that you can plot the data in WaveView. ‘nomod’ instructs the simulator to generate the simulation report, i.e. the ‘.lis’ file, without printing model details. You will find ‘nomod’ useful when there are multiple complex models within your netlist.

3. Run a sample low-pass filter circuit

- a. Simulate the circuit netlist you just created.

```
> hspice your_ckt.sp > your_ckt.lis
```

Upon proper completion of the simulation, you should see the following:

```
> info:          ***** hspice job concluded
```

- b. Definitions of the output files:

- a) your_ckt.sp is the input netlist (your file)
- b) your_ckt.sw0 is the DC sweep data output (used by awaves)
- c) your_ckt.tr0 is the transient data output (used by awaves)
- d) your_ckt.out is the output listing from HSPICE (look here for errors or text about the circuit)
- e) your_ckt.st0 is the simulation run information (usually not useful)
- f) your_ckt.ic0 is the information about the initial condition computed by HSPICE (usually don’t need to look at this)

4. View the results of the transient analysis

Load WaveView Analyzer and examine the results of the transient analysis by entering the following:

```
> awaves &
```

or

> **wv &**

Once WaveView loads, click on **File/Import Waveform File**. This will open a menu to select which output file to display. Your output *your_ckt.sw0* as well as *your_ckt.tr0* should be listed; if not, switch to the correct directory using the tab or the arrows. Once you have found your output files, double-click on *your_ckt.tr0*, which should import the transient data into **Output View** window.

To view the transient waveforms, click on the “+” sign in front of the file name “D0:your_ckt.tr0” and then double click on “*toplevel*”. Now you should see the waveform data listed below. To see the waveform, you can just double click on the waveform name and it will appear in a new panel on the right. In our case, we want to plot the waveforms for **in** and **out** in the same panel. To do this, select **v(vs)** by single clicking on it, then hold “Ctrl” and double click on **v(vo)**. You should see both waveforms appear on the same panel.

To save your trace data, press ‘**Ctrl+E**’.

5. DC operating point

- a. Create a netlist of the circuit in Figure T1.2. You may copy use the netlist shown below.

```
EE105 FALL18 SAMPLE DIODE CIRCUIT
.model diode_sample d (is=1e-14 vj=0.6 rs=10)
vs vs gnd 5V
r1 vs vo 5k
d1 vo gnd diode_sample
.op
.end
```

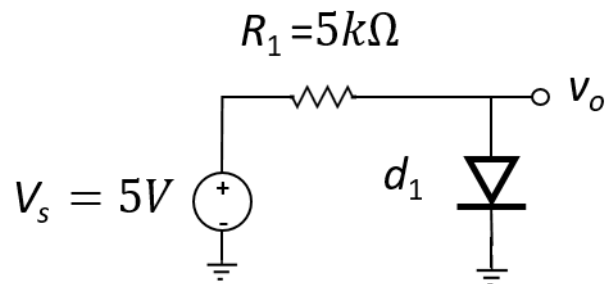


Figure T2.1

- b. Simulate this circuit and open the SPICE output file ‘filename.lis’ and scroll down until you see the following:

```
\***** operating point information ..... \*****'
.....
\+0:vo = 655.8104m 0:vs = 5.0000'
```

- c. From the output file we know that the DC output voltage is 655.8104 mV.

6. Syntax Reference

Any bracketed labels must be replaced entirely (i.e. if you want a value of 5 V, you should replace <value> with 5V).

- Independent voltage source

```
v<name> <+ terminal> <- terminal> <value>
```

- Independent current source

```
i<name> <+ terminal> <- terminal> <value>
```

- Voltage-controlled voltage source

```
E<name> <+ terminal> <- terminal> <+ control> <- control> <gain>
```

- Current-controlled voltage source (`vcontrol` refers to the voltage source which the controlling current flows through)

```
H<name> <+ terminal> <- terminal> <vcontrol> <gain>
```

- Voltage-controlled current source

```
G<name> <+ terminal> <- terminal> <+ control> <- control> <gain>
```

- Current-controlled current source (`vcontrol` refers to the voltage source which the controlling current flows through)

```
F<name> <+ terminal> <- terminal> <vcontrol> <gain>
```

- Sinusoidal source (used as a `<value>`)

```
sin(<offset> <amplitude> <frequency> <delay> <damping> <phase>)
```

- Square wave source (used as a `<value>`)

```
pulse(<vmin> <vmax> <delay> <rise time> <fall time> <pulse width> <period>)
```

- Piece-wise linear source (used as a `<value>`)

```
pwl(<t0> <v0> <t1> <v1> <t2> <v2> ...)
```

- Resistor

```
r<name> <terminal 1> <terminal 2> <value>
```

- Capacitor

```
c<name> <terminal 1> <terminal 2> <value>
```

- Inductor

```
l<name> <terminal 1> <terminal 2> <value>
```

- Model (type can be `nmos`, `pmos`, `NPN`, `PNP`, or `d` for diode)

```
.model <name> <type> (<parameter list>)
```

- MOSFET (you can specify additional parameters, such as `W=<value>` `L=<value>`, in the parameter list)

```
m<name> <drain> <gate> <source> <body> <model> <parameter list>
```

- BJT

```
q<name> <collector> <base> <emitter> <model> <parameter list>
```

- Diode

```
d<name> <+ terminal> <- terminal> <model> <parameter list>
```

- AC analysis (pick either lin, dec, or oct)

```
.ac <lin|dec|oct|> <number of samples> <freq start> <freq stop>
```

- DC analysis

```
.dc <source> <start> <stop> <step>
```

- Nested DC analysis (source1 is swept, source2 is stepped)

```
.dc <source1> <start1> <stop1> <step1> <source2> <start2> <stop2>  
<step2>
```

- Transient analysis

```
.tran <t step> <t stop>
```

- TF analysis

```
.tf v(<node>) <source>
```

- PZ analysis

```
.pz v(<node>) <source>
```