NGSPICE- Usage and Examples

Electronic Devices Lab : Lab 0

Department of Electrical Engineering
Indian Institute of Technology, Bombay
2023-07-31

Think it. Make it. Prove it.

What is NGSPICE?

- SPICE is an acronym for for **S**imulation **P**rogram with **I**ntegrated **C**ircuit **E**mphasis. First developed at UC Berkeley, it is the origin of most modern simulators.
- NGSPICE is an open source mixed-signal circuit simulator. It is the result of combining existing SPICE features with some extra analyses, modeling methods and device simulation features.
- It is freely available for use in Linux and Windows. It is recommended to use Linux for NGSPICE.
- NGSPICE requires you to describe your circuit as a **netlist**. A netlist is defined as a set of circuit components and their interconnections.

NGSPICE provides you with....

Basic Circuit Elements

- Passive components- resistors, capacitors, inductors, etc.
- Sources- independent voltage and current sources, controlled sources

Semiconductor Devices

- Pre-defined circuit elements such as diodes and transistors
- Allows you to define or include models of specific devices e.g. specialized transistors and op-amps

Circuit Analysis Techniques

- DC and AC analyses
- Transient and Steady-state analyses
- Pole-Zero, Noise analyses and more

Think it. Make it. Prove it.

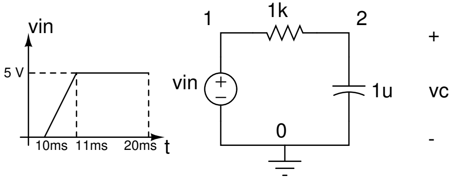
Getting Started with NGSPICE

- You can use any text editor (gedit or notepad++) to write your circuit netlist. The first line in an NGSPICE file is **not executed**. It is used to describe the aim of the circuit being simulated.
- All NGSPICE comments start with an asterix, i.e. ‘*’
- The NGSPICE file comprises of the circuit netlist followed by the details of the analysis the user wishes to do.
- NGSPICE files are usually saved with the extension ‘.cir’ or ‘.spice’.
- Circuit components are identified by their first letter of naming, called **prefix**, i.e. resistors begin with `r`, capacitors with `c`, bipolar transistors with `q`, MOSFETs with `m`, voltage sources with `v` and so on.
- All circuit nodes are named/numbered. The netlist requires one ground node (zero potential).

Think it. Make it. Prove it.

Example I- Transient Analysis of an RC Circuit

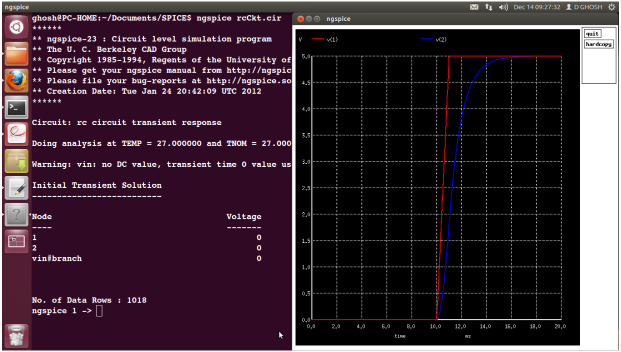
A simple RC circuit, excited by a user-defined signal V_{in}. We want to find the capacitor voltage i.e. $V_c(t)$. The netlist is as shown.



```
RC Circuit Transient Response
*resistor connected between nodes 1 and 2
r1 1 2 1k
*capacitor connected between nodes 2 and 0
c1 2 0 1u
*piecewise linear input voltage
vin 1 0 pwl (0 0 10ms 0 11ms 5v 20ms 5v)
*transient analysis for 20ms, step size 0.02ms
.tran 0.02ms 20ms
*defining the run-time control functions
.control
run
*plotting input and output voltages
plot v(1) v(2)
.endc
.end
```

Example I- Transient Analysis of an RC Circuit (cont'd)

- Once you have typed your netlist, save it with an appropriate name, say `rcCkt.cir`.
- Open the Linux terminal, and change the working directory to the folder where your netlist file is saved. e.g. if the file is in the `Documents` folder, type `cd ∼/Documents` in the the command prompt.
- Run the netlist file using the command `ngspice rcCkt.cir`
- And that's it- your netlist should run! A snapshot is as shown.



```
ghosh@PC-HOME:~/Documents/SPICE$ ngspice rcCkt.cir
******
** ngspice-23 : Circuit level simulation program
** The U. C. Berkeley CAD Group
** Copyright 1985-1994, Regents of the University of
** Please get your ngspice manual from http://ngspic
** Please file your bug-reports at http://ngspice.so
** Creation Date: Tue Jan 24 20:42:09 UTC 2012
******

Circuit: rc circuit transient response

Doing analysis at TEMP = 27.000000 and TNOM = 27.000

Warning: vin: no DC value, transient time 0 value us

Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
1                                            0
2                                            0
vin#branch                                   0



No. of Data Rows : 1018
ngspice 1 ->
```

Think it. Make it. Prove it.

Inside the NGSPICE Shell

- Once any netlist is run by NGSPICE, the terminal is hooked to an NGSPICE shell, with a prompt such as `ngspice 1 ->`. You can exit the NGSPICE shell anytime by typing `exit` or `quit`.
- It is not always necessary to quit NGSPICE every time to run a new netlist. You can use the command `source <filename>.cir` to simulate a new netlist file using the NGSPICE prompt.
- The commands specified between `.control` and `.endc` in the netlist file may be used in the NGSPICE prompt separately.
- The waveforms may be saved as a postscript file by clicking on the hardcopy icon on the waveform window. However, this saves it as a default filename in the root directory. A better way to save the waveform in the working directory would be to use the following commands (say for the R-C circuit discussed above)

```
set hcopydevtype=postscript
hardcopy rcPlot.ps v(1) v(2)
```

- You can even save the plots using different colours. Read the NGSPICE manual for that, and much more!

Think it. Make it. Prove it.

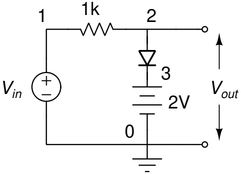
Example II- AC Analysis of RC Circuit

For the same R-C circuit discussed in Example I, let us do the small-signal AC analysis, i.e. find its frequency response.After running this, you should be able to see two plot windows- a magnitude (dB) plot and a phase (degrees) plot.

```
RC Circuit Frequency Response
r1 1 2 1k
c1 2 0 1u
*Specifying an AC source with zero dc
vin 1 0 dc 0 ac 1
*AC analysis for 1 Hz to 1MHz, 10 points per decade
.ac dec 10 1 1Meg
.control
run
*Magnitude dB plot for v(2) on log scale
plot vdb(2) xlog
*Phase degrees plot for v(2) on log scale
plot {57.29*vp(2)} xlog
.endc
.end
```

Example III- DC Analysis of a Shunt Clipper

A DC analysis involves varying a voltage or a current source output throughout a range of values. Consider the following shunt clipper circuit. We wish to find the V_{out} v/s V_{in} characteristic for this circuit, say for -5 V $\leq V_{in} \leq$ +5 V.



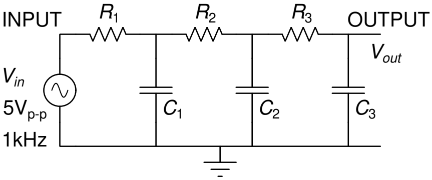
```
Shunt Clipper DC analysis
r1 1 2 1k
*Specifying a default diode p n
d1 2 3
*Independent DC source of 2V
vdc 3 0 dc 2
*Independent DC source whose voltage is to be varied
vin 1 0 dc 0
*DC Analysis on source vin, to vary from -5 to +5V
.dc vin -5 5 0.01
.control
run
plot v(2) vs v(1)
.endc
.end
```

Use of Subcircuits

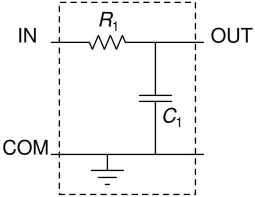
- We have seen how to use electronic devices to build a circuit and test it using NGSPICE.
- Various electronic devices have their own existing model files that represent the electrical behaviour of that device, which we can use in a netlist. What if we now have an existing circuit, and want to use it to build bigger circuits?
- A typical example is using an op-amp (operational amplifier) to design a simple amplifier or a filter. Note that, an op-amp is a pre-existing **circuit** and not a device. It is made of many transistors.
- NGSPICE allows us to define an op-amp as a **subcircuit**. A subcircuit is much like an IC- we know its pins to interface with the outside world, but we need not be familiar with the inside circuit!
- A subcircuit is a collection of devices familiar to SPICE. A subcircuit is identified by the prefix `x`.
  The usage is very similar to that of a model file.

An Example Scenario for Subcircuits

Consider the circuit shown (each resistor is 1 kΩ and each capacitor 1 μF)



- If we were to write this netlist, there would be a number of useless reiterations of resistors/capacitors.
- Since the component values are repetitive, we can define each $R - C$ pair as a subcircuit, with terminals `IN`, `OUT` and `COM`.



Multiple *R*-*C* Networks using Subcircuits

The netlist to do a transient analysis for the above circuit is as follows

```
RC Circuit Transient Analysis using Subcircuits
*Defining the Subcircuit RC_subcircuit:  <name> <pin1> <pin2> <pin3>
.subckt RC_subcircuit IN OUT COM
r IN OUT 1K
c OUT com 1u
.ends
*Subcircuit definition ends here
vsin INPUT gnd sin(0 2.5 1K 0 0)
*Invoking RC subcircuit with an ’x’
xrc1 INPUT 1 gnd RC_subcircuit
xrc2 1 2 gnd RC_subcircuit
xrc3 2 OUTPUT gnd RC_subcircuit
.control
tran 0.02m 40m
plot v(INPUT)
plot v(OUTPUT)
.endc
.end
```

Think it. Make it. Prove it.

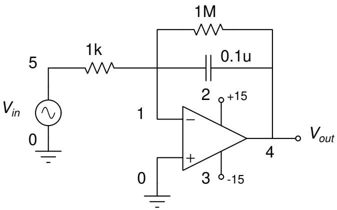
Multiple *R*-*C* Networks using Subcircuits (cont'd..)

In continuation, a subcircuit can be invoked, and different component values can be assigned while invoking (similar to parameter passing). Let us see how!

```
RC Circuit Transient Analysis using Subcircuits
*Providing some default component values
.subckt RC_subcircuit IN OUT COM r1 = 1k c1 = 1u
r IN OUT {r1}
c OUT COM {c1}
.ends
vsin INPUT gnd sin(0 2.5 1k 0 0)
xrc1 INPUT 1 gnd RC_subcircuit
*Instantiating component values for 2nd subcircuit
xrc2 1 2 gnd RC_subcircuit r1 = 100 c1 = 0.1u
xrc3 2 OUTPUT gnd RC_subcircuit
.control
tran 0.02m 40m
plot v(INPUT) v(OUTPUT)
.endc
.end
```

The quantities in {} are parameters, treated as mathematical quantities. Apart from `xrc2`, all other subcircuits have the default `r1` and `c1` values.

Example- Integrator using op-amp (741)



```
Integrator using op-amp 741
*Including the predefined op-amp subcircuit file
.include ua741.txt
*Connections as mentioned in subcircuit file
x1 0 1 2 3 4 UA741
r1 5 1 1k
c1 4 1 0.1u
rf 4 1 1Meg
vcc 2 0 dc 15v
vee 3 0 dc -15v
*Giving a sinusoidal input
vin 5 0 sin (0 1v 1k 0 0)
.tran 0.02ms 6ms
.control
run
plot v(5) v(4)
.endc
.end
```

Exercises

- Consider the shunt clipper shown in Example III. Change (a) the diode connections (b) the 2 V battery polarity and simulate to observe V_{out} v/s V_{in} variation. For each case, give a 1 kHz sinusoidal input and observe the output waveform (six cycles).
- Write an NGSPICE netlist for a diode-based bridge rectifier and simulate it to observe the rectified voltage across a load resistor, by giving a 12 V, 50 Hz input. Also, observe the V_{out} v/s V_{in} transfer characteristics.